

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



Mobile Web Accessibility Evaluation

António Rogério Crisóstomo de Campos Bandeira

MESTRADO EM INFORMÁTICA

2010

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



Mobile Web Accessibility Evaluation

António Rogério Crisóstomo de Campos Bandeira

DISSERTAÇÃO

Projecto orientado pelo Prof. Doutor Luís Manuel Pinto da Rocha Afonso Carriço

MESTRADO EM INFORMÁTICA

2010

Abstract

This work presents an approach to mobile Web accessibility evaluation allowing for Web content accessibility and mobile adequacy assessment, regarding different selectable disability profiles. We propose an extension to existing methodologies in order to fully encompass a thorough integration of mobile and accessibility guidelines within specific impairments perspectives and a prototype that serves as a proof of concept tool for that methodology.

The integrated guidelines include MWBP and WCAG recommendations. It is well known that these sets overlap. Some subsets of MWBP should always be considered regardless of the users' special needs, other are relevant for specific disability types. We claim that when specific disabilities are addressed some MWBP might become irrelevant.

We developed a proof of concept tool, MWAAT, which fully addressed the basic concepts of the proposed methodology. Its analysis, design and the most relevant implementation aspects are presented in this dissertation. Regarding the development perspective, MWAAT adopts a very flexible approach envisioning its future integration into the ACCESSIBLE project platform and its potential extension into other guideline sets and alternative scenarios.

As other available tools, it offers mechanisms for WCAG assessment or MWBP evaluation in an independent manner. However, unlike any other, it permits the evaluation of Web content through an adequate combination of both guidelines sets, as a whole, or more importantly, for a specific disability. As such, it provides a powerful mechanism to Web content developers and most notably to consolidate the selective disability assessment approach.

This dissertation illustrates these capabilities through representative case studies. The results herein shown reveal not only the assessment differences between desktop and mobile content, but also the difference between these at the impairment level.

Keywords: Mobile Web, Accessibility, Methodology, Evaluation, Assessment Tool

Resumo alargado

A World Wide Web oferece uma vasta quantidade de informação e serviços e o seu potencial para melhorar a vida das pessoas e elevar o seu padrão de vida é enorme. A disciplina de acessibilidade da Web procura permitir que as pessoas com deficiência possam utilizar a Web tal como todas as outras, sem barreiras de acesso aos seus conteúdos. Tornar os conteúdos Web acessíveis às pessoas com deficiência é parte integrante de sítios da Web de elevada qualidade, constitui uma oportunidade de mercado emergente e, num número crescente de casos, constitui uma exigência legal como por exemplo a resultante da Secção 508 nos E.U.A.

Os dispositivos móveis são cada vez mais usados como terminais de acesso à Internet. No entanto, as características intrínsecas e as limitações destes dispositivos podem colocar obstáculos na interação com a Web. Convém ainda salientar que para além das questões específicas à interação através de dispositivos móveis, pessoas com deficiência também podem aceder à Web a partir desses dispositivos colocando-se então questões adicionais de acessibilidade.

Muitos designers e técnicos que desenvolvem aplicações para a Web móvel não estão familiarizados com as peculiaridades destes dois mundos. Existem diferentes conjuntos de directrizes para o desenvolvimento de conteúdos Web e para o desenvolvimento de conteúdos adequados para dispositivos móveis. As recomendações Web Content Accessibility Guidelines (WCAG) por exemplo, definem conjuntos de regras para tornar os conteúdos Web acessíveis a pessoas com deficiência, enquanto que as recomendações das Mobile Web Best Practices (MWBP), definem regras para tornar os sítios Web mais adequados ao seu acesso a partir de dispositivos móveis. Felizmente existe uma sobreposição, embora parcial, entre elas.

A evolução contínua em ambas as áreas, derivada quer da publicação de novas orientações de acessibilidade quer pela evolução e crescente diversidade de dispositivos móveis, dificulta o desenvolvimento de aplicação e conteúdos que sigam as diferentes directivas. Se tomarmos ainda em consideração diferentes tipos de deficiência e as respectivas limitações daí derivadas para a interação, as dimensões do quebra-cabeça tornam-se ainda mais complexas. Na verdade, quem desenvolve pode ter que entrar em consideração com orientações de acessibilidade e orientações para o desenvolvimento em plataformas móveis, para além de ter que considerar utilizadores com diferentes tipos de deficiência das quais

resultam diferentes potenciais barreiras de interacção. Em suma, não constitui uma tarefa fácil.

Para superar todos estes diferentes aspectos, quem desenvolve precisa de suporte durante o processo de desenvolvimento ao longo das várias etapas do ciclo de vida das aplicações. Existem várias ferramentas para a avaliação de acessibilidade de sítios Web ou para avaliação da sua adequação para serem acedidos partir de dispositivos móveis. Em geral, porém, elas suportam-se numa abordagem onde todas as orientações são testadas indiferentemente não tomando em consideração as especificidades relativas aos utilizadores, aos dispositivos de destino e aos conjuntos de restrições de mobilidade e acessibilidade relevantes daí resultantes. Mesmo que em trabalhos recentes se abordem algumas destas nuances, a verdade é que uma abordagem global onde se tome em consideração as especificidades relativas aos diferentes cenários resultantes de utilizadores com diferentes tipos de deficiência acedendo aos sítios Web através de diferentes tipos de dispositivos, nomeadamente dispositivos móveis, ainda não existe.

No trabalho relacionado desta dissertação, introduzem-se os conceitos e principais questões relativas à acessibilidade dos conteúdos Web, para diferentes tipos de deficiência, e as características e requisitos específicos do seu acesso a partir de dispositivos móveis. Seguidamente apresentam-se as principais recomendações existentes no âmbito da acessibilidade e as principais recomendações existentes no domínio da adequação dos conteúdos Web para o acesso a partir de dispositivos móveis. Discutem-se também as ferramentas existentes e apresentam-se lacunas existentes. O trabalho relacionado termina apresentando o projecto ACCESSIBLE em cujo âmbito o trabalho desta tese foi efectuado e a metodologia de avaliação de acessibilidade, Accessible Harmonized Methodology (HAM), efectuada no âmbito do mesmo.

Nesta dissertação apresenta-se uma abordagem para a avaliação de acessibilidade de conteúdos Web para dispositivos móveis. A abordagem permite a avaliação de acessibilidade e de adequação para acesso a partir de dispositivos móveis, de conteúdo Web, para diferentes perfis de deficiência seleccionáveis. Isto irá permitir a quem desenvolve conteúdos para a Web, aos designers e aos especialistas em avaliação, efectuar rapidamente avaliações especializadas tendo em consideração diferentes tipos de deficiência e diferentes contextos de entrega dos conteúdos. Neste trabalho apresentamos também a ferramenta desenvolvida para permitir avaliar a abordagem.

Começa-se por analisar a introdução da dimensão móvel no processo de avaliação de acessibilidade sendo considerados diferentes perfis de deficiência nessa avaliação. A identificação do subconjunto de orientações relevante que deve ser aplicado a cada perfil é escrutinado, tendo em consideração as recomendações das MWBP, as recomendações das WCAG e a abordagem da HAM.

Dessa análise do conjunto de diretrizes WCAG, e considerando o contexto em que se aplica, chega-se à conclusão de que os pontos de verificação de acessibilidade que

são relevantes para um tipo de deficiência quando se usa um desktop não muda para um ambiente móvel. Em vez disso a sua relevância tende a ser reforçada pela relação referida entre as recomendações WCAG e MWBP.

Em relação às MWBP, três subconjuntos de orientações devem ser consideradas, nomeadamente:

- Orientações directamente relevantes para um ou mais tipos de deficiência;
- Orientações relevantes para o acesso através de dispositivo móvel, independentemente das necessidades dos utilizadores especiais;
- Orientações que se tornam irrelevantes para alguns tipos de deficiência no acesso a partir de dispositivos móveis

O primeiro subconjunto, deriva da relação entre as orientações MWBP e WCAG. A correspondência entre as orientações MWBP e as limitações de interação dos diferentes tipos de deficiência podem ser directamente derivados a partir de documentação da W3C.

O significado desse subconjunto deve porém ser cuidadosamente avaliado. Em primeiro lugar, a relação entre as orientações das WCAG e das MWBP, nem sempre correspondem a uma equivalência completa ou mesmo a uma implicação. Isso significa que, na maioria das vezes, ambas devem ser avaliadas quer em termos de eventual falha de verificação de conformidade com a orientação, quer em termos das situações de alerta ou erro verificadas e das mensagens correspondentes.

O facto de existir uma relação entre algumas das orientações, não exclui a necessidade de verificar as restantes orientações MWBP não relacionadas com as orientações WCAG. Há aspectos importantes para a adequação do conteúdo a ser acedido por um dispositivo móvel que não dizem directamente respeito a questões específicas de acessibilidade. As orientações quanto às características, tais como codificação de caracteres, content format preferred, formatos de conteúdo desejado, cookies, etc., não têm uma relação com um problema específico de acessibilidade, mas são essenciais para a interação de dispositivos móveis em geral.

Considere-se, por exemplo, um tipo de deficiência visual e/ou um utilizador que, por regra, inibe o download de imagens no agente de navegação do seu dispositivo móvel. Aplicando testes relacionados com as orientação MWBP de conformidade de imagens (por exemplo, especificação explícita do tamanho de imagens) pode resultar em detecção de falhas irrelevantes para esse tipo de utilizador. Na verdade, não ter a especificação do tamanho da imagem não muda em nada a experiência deste tipo de utilizador uma vez que a imagem não será descarregada de qualquer forma.

Tal como no exemplo anterior, outros casos devem ser tomadas em consideração para evitar resultados de avaliação que dêem origem a falsos positivos. Assim, uma ferramenta ou uma metodologia de avaliação da acessibilidade de conteúdos para a Web móvel,

que suporta diferentes tipos de deficiência, deve definitivamente fazer referência a estas orientações das quais resultam falsos positivos. Na melhor das hipóteses, eles devem ser tratados como casos particulares, provavelmente associados a baixo nível de severidade, ou simplesmente removido dos conjuntos de recomendações e testes relevantes para a deficiência específica.

Após a apresentação da aproximação metodológica, nesta dissertação apresenta-se a especificação de requisitos da ferramenta MWAAT (Mobile Web Accessibility Assessment Tool), utilizando diagramas de casos de uso para a sua descrição. Apresentam-se ainda os diagramas de classes e os requisitos não-funcionais.

São apresentadas as principais considerações de design da ferramenta MWAAT. A arquitectura é descrita, sendo ainda o apresentados os diagramas de interação dos casos de uso mais relevantes. Apresentam-se ainda os diagramas de classes do sistema.

A nível da implementação são apontados os aspectos mais relevantes, tais como o ambiente em que foi desenvolvido e os aspectos mais relevantes de implementação dos principais módulos da arquitectura da ferramenta, nomeadamente o seu interface gráfico, o componente de selecção de cenários de utilização, o componente de manipulação e apresentação de resultados e o componente de avaliação considerando quer os mecanismos de acesso e manipulação dos recurso Web sejam eles URL ou ficheiros, quer os mecanismos de avaliação e os componentes de análise e testes implementados.

Finalmente nesta dissertação são apresentados resultados referentes a um cenário de desenvolvimento de um conteúdo Web e três casos de estudo de avaliação de conteúdos Web existentes que ilustram a utilização e potencial da metodologia e da ferramenta MWAAT. No primeiro caso de estudo, a ferramenta MWAAT é utilizada para ajudar criar um conteúdo acessível para a Web móvel, a partir de um ficheiro HTML ainda não instalado num Web Server. Este exemplo serve principalmente para demonstrar o uso da ferramenta.

Nos outros três casos de estudo acedem-se a diferentes recursos disponiveis na Web, simulando o acesso a partir de dois contextos de acesso diferentes, o acesso a partir de um contexto desktop e o acesso a partir de um contexto móvel. As diferentes representações dos conteúdo Web acedidos, recebidas para os diferentes contextos de acesso foram avaliados segundo diferentes cenários de avaliação nomeadamente a avaliação de acessibilidade standard sem considerar nenhum tipo específico de deficiência e a avaliação de acessibilidade considerando tipos especificos de deficiência nomeadamente, deficiência visual, deficiência auditiva, daltonismos e deficiência motora. Foram ainda efectuados testes para avaliar a adequação de conteúdos para serem acedidos em dispositivos móveis, sendo testada a acessibilidade sem considerar nenhum tipo especifico de deficiência e a acessibilidade considerando os diferentes tipos de deficiência anteriormente referidos.

Os resultados obtidos permitem concluir como válidos alguns pressupostos desta dissertação nomeadamente revela-se claro que para as deficiências específicas, os conteúdos têm

muito menos problemas de acessibilidade do que quando se avalia o caso geral indiscriminado, uma vez que cada conjunto de orientações relevantes para cada deficiência é um subconjunto dos testes disponíveis. Uma análise mais profunda dos resultados da avaliação mostraram que mesmo quando os números são semelhantes entre os diferentes tipos de deficiência, os problemas reais levantados correspondem geralmente a diferentes orientações que não são observadas.

Isto reforça a decisão de ter um conjunto de testes específicos para cada tipo de deficiência, uma vez que, por exemplo, para o caso deficientes auditivos alguns dos sítios Web avaliados são totalmente acessíveis. Olhando para a dimensão mobilidade, fica claro que a representação móvel apresenta um tamanho muito menor do que a representação padrão em todos os casos de estudo. Isso ocorre do facto de estes sítios Web terem uma representação específica para ser acedida a partir de contextos móveis, que geralmente oferece uma versão simplificada que é mais adequada.

Em relação ao números absolutos, o ganho em termos de acessibilidade é enorme quando se compara a representação móvel com a representação padrão, tanto em termos de número de nós como em número de advertências e de erros. Olhando para as percentagens, o mesmo é verdade para a maioria dos casos de estudo.

Em geral, a melhoria verificada na acessibilidade das representações móveis versus as representações padrão está em conformidade com o esperado. Em todos os casos de estudo as diferenças entre a avaliação de acessibilidade geral e a avaliação para deficiências específicas são visíveis.

A presente dissertação termina sintetizando as conclusões derivadas dos resultados obtidos e expondo o trabalho futuro previsto, referente à extensão do conjunto de testes disponíveis, à evolução do interface com o utilizador, à evolução prevista para o manuseamento e apresentação de resultados, aos casos de experimentação e ao trabalho de investigação previsto para a refinação dos cenários de avaliação.

Palavras-chave: Web Móvel, Acessibilidade, Metodologia, Avaliação, Ferramenta

Contents

Table of contents	xi
Lista de Figuras	xvi
Lista de Tabelas	xix
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	2
1.3 Institutional Context	3
1.4 Publications	3
1.5 Document Outline	3
2 Related Work	5
2.1 Disabled and mobile use and barriers	5
2.2 Web Content Accessibility Standards	6
2.2.1 WCAG 1.0	6
2.2.2 WCAG 2.0	7
2.2.3 Section 508	8
2.3 Mobile Web Content Standards	8
2.3.1 Mobile Web Best Practices	9
2.3.2 Mobile OK basic tests	10
2.4 Assessment Tools	11
2.4.1 Accessibility assessment tools	11
2.4.2 Mobile adequacy assessment tools	12
2.5 Accessibility and mobile adequacy assessment	13
2.6 The ACCESSIBLE Project	14
2.6.1 The Harmonized Accessible Methodology	17
2.7 Summary	19
3 Analysis	21
3.1 Mobile Web Accessibility for Disability Profiles	21

3.1.1	WCAG checkpoints on a mobile usage setting	21
3.1.2	MWBP for disability types	22
3.2	Requirements Specification	24
3.2.1	Use Cases	25
3.2.2	Use Cases Definition	27
3.3	Class diagrams	34
3.4	Non-functional requirements	36
3.5	Summary	37
4	Design	39
4.1	Architecture	39
4.1.1	MWAAT	39
4.1.2	User Interface	40
4.1.3	Selector	40
4.1.4	Evaluator	41
4.1.5	Results Handler	41
4.2	From requirements to design	42
4.2.1	Use Cases Design	42
4.2.2	Design Class Diagrams	50
4.3	Summary	51
5	Implementation	53
5.1	Environment	53
5.2	Graphical User Interface	53
5.3	MWAAT	56
5.4	Selector	56
5.5	Results handler	57
5.6	Evaluator	57
5.6.1	Negotiating resource representations	58
5.6.2	Parsing HTML	58
5.6.3	Executing the assessment tests	59
5.7	Summary	61
6	Results	63
6.1	Web content development example	64
6.2	Case studies	67
6.2.1	Web Portal	67
6.2.2	Web Magazine	72
6.2.3	Web Finance Portal	73
6.3	Aggregated Results	73

6.4	Summary	77
7	Conclusions and Future Work	79
7.1	Future Work	80
7.1.1	Set of Available Tests	80
7.1.2	User Interface	81
7.1.3	Results Handler	81
7.1.4	Selector Component and Inference Engine	81
7.1.5	Further Validation of Methodology and Tool	82
7.1.6	Refining Assessment Scenarios	82
	Abreviaturas	85
	Bibliografia	89
	Índice	90

List of Figures

2.1	Overall Architecture of ACCESSIBLE System	15
2.2	HAM Approach	18
2.3	Disability type associated Interaction Limitations example	18
2.4	Example of best practice catalogue for Web Applications	19
3.1	Mapping Disability Type to MWBP	22
3.2	Use Case diagram for MWAAAT	26
3.3	MWAAT Main Class Diagram	35
3.4	Tests Class Diagram	36
4.1	MWAAT Block Diagram	40
4.2	Select relevant guidelines	42
4.3	Disability type and device type initialization	43
4.4	Set proxy	44
4.5	Select delivery context	45
4.6	Handle results	46
4.7	Results default filename setting	47
4.8	Assess URL - resource handling and parsing	48
4.9	Assess URL - test evaluation	49
4.10	Design Class Diagram	50
4.11	Mobile adequacy tests class diagram	51
5.1	MWAAT graphical user interface	54
5.2	Save Results user interface	55
5.3	Evaluate HTML GUI	55
5.4	Blind disability associated guidelines	56
5.5	Motor impaired associated guidelines	56
5.6	Mobile content guidelines	57
6.1	Web Content Example	64
6.2	Mobile devices content adequacy	65
6.3	Mobile devices accessibility for motor impaired users	65
6.4	Mobile devices accessibility for blind users	66

6.5	Mobile devices content adequacy and listing source code	66
6.6	Sapo Home Page	67
6.7	Default representation accessibility evaluation	68
6.8	Mobile representation mobile adequacy evaluation	68
6.9	Mobile accessibility evaluation	69
6.10	Mobile accessibility evaluation for blind disability	70
6.11	Mobile accessibility evaluation for deaf disability	70
6.12	Mobile accessibility evaluation for color blind disability	71
6.13	Mobile accessibility evaluation for motor impaired disability	72
6.14	Engadget Home Page rendered on a desktop browser	72
6.15	Yahoo Finance home page rendered on a desktop browser	73

List of Tables

- 3.1 Example of relevant checkpoints for disability type - a cell with x means that the checkpoint is relevant for that disability type 22
- 3.2 Relevant MWBP guidelines for disability type - a cell with x means that the checkpoint is relevant for that disability type; an empty cell means that relevance is minimal or absent 24
- 5.1 HTML node content query examples 60
- 5.2 An evaluation example 60
- 6.1 Case Study - Web Portal Accessibility Evaluation 74
- 6.2 Case Study - Web Magazine Accessibility Evaluation 75
- 6.3 Case Study - Web Finance Portal Accessibility Evaluation 76

Chapter 1

Introduction

1.1 Motivation

The World Wide Web provides a wealth of information and services and its potential to improve people's lives and raise their standard of living is enormous. The Web accessibility discipline strives about enabling people with disabilities to use the Web just like the unimpaired, without barriers. Making Web sites accessible for people with disabilities is an integral part of high quality Web sites, a growing market opportunity and, in a growing number of cases a legal requirement (e.g., following Section 508 [6] in the USA).

At the same time, we are being faced with an explosion in mobile devices usage all over the world (including the developing world). Mobile devices are increasingly being used as a terminal to access the Web, its information and services. However, the intrinsic features and limitations of mobile devices are a hinder to Web interaction. Additionally to mobile specific constraints, people with disabilities might also access the Web from mobiles devices.

Many Web designers and mobile application developers are not familiar with the peculiarities of these two worlds, and different sets of guidelines to develop accessible Web contents and mobile-friendly Web contents exist. The Web Content Accessibility Guidelines (WCAG) [11], [9] defines a set of rules to make Web sites accessible to people with disabilities, whereas the Mobile Web Best Practices (MWBP) [28] define rules for making Web sites more usable from a mobile device. Interestingly, there is a recognized partial overlap between them [13].

Nevertheless, the continuous evolution in both areas, whereas by the publication of new guidelines or by the evolution and increasing diversity of mobile devices, complicates the development of applications that follow both directives. Moreover, if we take into account different types of disabilities and their inherent distinct usage and accessibility constraints, the dimensions of the puzzle becomes even more intricate. In fact developers may have to consider, evolving accessibility and mobile guidelines and different characteristics of disabled users and of mobile devices. Overall, it is not an easy

task.

To overcome all of these aspects, developers need to be assisted during development processes in several steps of their applications development life cycle. Several tools are already available for the assessment of Web sites, in terms of their accessibility [1], [18], [36], and in terms of their mobile usage [15], [29]. In general though, they tend to adopt brute force approaches where all the guidelines are applied indifferently of the target users, the target devices or the conjunction of mobile and accessibility constraints. Even if recent work [13] is emerging that addresses some of these nuances, the fact remains that an overall comprehensive approach is still lacking, both in terms of an articulated framework or a full understanding of the intersections of the evolution and differentiation dimensions.

This work presents an approach to mobile Web accessibility evaluation. It allows for Web content accessibility and mobile adequacy evaluation, regarding different selectable disability profiles. This will provide the necessary support to Web developers, designers and assessment experts to conduct rapid, yet specialized, accessibility assessments focused on different disability types for Web sites tailored also to mobile devices. We also propose a prototype that, integrated with the remaining system, will be used as a proof-of-concept of the approach.

1.2 Contributions

The contributions of this thesis include:

- At the methodological level we analyze the integration between knowledge concerning Accessibility design and Mobile design, and introduce a third dimension on personalized assessment regarding different disability profiles. As a consequence we raise the awareness to the fact that this third aspect precludes the simplistic approach of a straight guidelines combination, which considers only its overlapping. The choice of the adequate mobile adequacy guidelines rules is also directly dependent on the specific disability profile.
- At the technological level, a tool was developed that serves as a proof of concept for the above mentioned methodological contribution. The tool includes the ability to choose the disability profile and the device support that will render the Web content, and select accordingly the adequate subset of guidelines to be applied. In top of that, also supports different context delivery requests simulation, namely the mobile and default Web content representations assessments.

The tool was used in a set of case studies that illustrate the approach of evaluating the accessibility of mobile Web content, within specific disability perspectives. The analy-

sis shows interesting results that pave the way to continue the research in order to fully understand and validate refinement of the assessment profiles.

1.3 Institutional Context

The present work took place at the Large-Scale Informatics System Laboratory (LaSIGE), a research unit of the Informatics Department of the University of Lisbon, Faculty of Sciences.

It was developed within the scope of the ACCESSIBLE project, partially funded by the EC FP7 project ACCESSIBLE - Accessibility Assessment Simulation Environment for New Applications Design and Development, Grant Agreement No. 224145.

1.4 Publications

With the goal of disseminating, validating and improve the current work, this thesis generated the following refereed publication:

- Rogério Bandeira, Rui Lopes and Luís Carriço, “Towards mobile Web accessibility evaluation”, in Proceedings of the ETAPS 2010 FOSS-AMA Workshop (ETAPS 2010), Paphos, Cyprus, March 2010.
- Rui Lopes, Rogério Bandeira, Luís Carriço and Karel Van Isacker, Towards mobile Web accessibility, vision and challenges, in Proceedings of the 1st International AEGIS Conference, Seville, Spain, October 2010, accepted to be published.

1.5 Document Outline

The remainder of this document is structured as follows:

Chapter 2 Related Work, introduces disabled and mobile use and the barriers faced.

It presents accessibility and mobile Web content standards and existing accessibility and mobile adequacy tools are discussed as well as their existing gaps. The ACCESSIBLE project scope within which this thesis’ work in undertaken is also described.

Chapter 3 Analysis, where the application domain problem is further scrutinized, considering the articulation of mobile adequacy and accessibility evaluation. The chapter also presents the functional requirements through use cases. Class diagrams and a set of non-functional requirements are also described.

Chapter 4 Design, presents the tool architecture, the interaction diagrams for the most relevant use cases and the design class diagram.

Chapter 5 Implementation, depicts the development environment and the most relevant aspects of each of the MWAAT' modules implementation.

Chapter 5 Results, where an introductory example of MWATT's usage is described showing its application to the development of mobile Web accessible content. Three URL case studies evaluations are also analyzed.

Chapter 6 Conclusion and Future Work, summarizes the dissertation main achievements and presents some of the most interesting directions to pursue in future work.

Chapter 2

Related Work

2.1 Disabled and mobile use and barriers

Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web as effectively as all the others [17]. Web accessibility is also important for older people whose abilities change due to aging [17]. Moreover users of mobile devices and people with disabilities experience similar barriers when interacting with Web content [31].

There are several standardization bodies concerned with the emergence of accessible Web contents and applications like the *Barrierefreie Informationstechnik-Verordnung* (BITV) the German Federal Ordinance on Barrier-Free Information Technology [5], the United States Federal Agency Section 508 Coordinators (section 508) [6], the Japanese Industrial Standards committee (JISC) and of course the World Wide Web Consortium (W3C), the main international standards organization for the World Wide Web, among others.

Worldwide there are accessibility laws and policies. Many countries adopted the World Wide Web Consortium (W3C) Web Content Accessibility Guidelines (WCAG) 1.0 or a variation containing also rules from the local jurisdiction [30]. At least 25 countries including Portugal have Web design laws and policies. European countries belonging to this group include Austria, Denmark, Finland, France, Germany, Ireland, Italy, The Netherlands, Norway, Portugal Spain, Sweden and the European Union itself.

W3C provides guidelines on making accessible Web content and best practices for mobile-friendly content and there is an overlap between them both [31].

WCAG and MWBP both aim to improve the Web interaction experience of users eliminating the barriers resulting from their disabilities or the device characteristics and limitations [13].

While there is an overlapping between the two guidelines in many areas, WCAG has requirements that are specific to accessibility needs of people with disabilities, and that are not relevant at all for mobile devices and MWBP has requirements that are specific to

mobile devices only regardless of the user capabilities.

Following these two guidelines makes the Web content more accessible to everyone regardless of the usage situation, environment, or device and designing to the guidelines together, instead of separately, can make the process more efficient.

In these continually evolving heterogeneous environment, adherence to standards is the only solution to common language.

2.2 Web Content Accessibility Standards

Web accessibility depends not only on accessible content but also on accessible Web browsers and other user agents. How these components of Web development and interaction work together is described in the essential components of Web Accessibility [16]. W3C also issued recommendations regarding User Agent Accessibility Guidelines (UAAG) [19] and Authoring Tool Accessibility Guidelines (ATAG) [32].

Regarding Web content accessibility, the Web Content Accessibility Guidelines from the World Wide Web Consortium are the closest we can get to official accessibility standards [11], [9].

2.2.1 WCAG 1.0

The Web Content Accessibility Guidelines from the World Wide Web Consortium explain how to make Web Content accessible to people with disabilities. They are intended to help Web content developers (authors, and designers).

The primary goal of these guidelines is to promote accessibility but following this recommendations also help to make Web content more available to all users whatever user agent they are using (e.g., desktop browser, voice browser, mobile phone, automobile-based personal computer, etc.) or constraints they may be operating under (e.g., noisy surroundings, under- or over-illuminated rooms, in a hands-free environment, etc.).

One important aspect to retain is that these guidelines do not discourage content developers from using multimedia content but rather explain how to make it accessible to a wider audience and in a wider range of usage situations.

By following WCAG 1.0 guidelines, content developers can create pages that transform gracefully remaining accessible despite constraints such as physical, sensory, and cognitive disabilities, work constraints, and technological barriers.

According to WCAG designing pages that transform gracefully [11]: follow some keys such as:

- Separate the structure from the presentation
- Provide text and text equivalents since text can be rendered in ways that are available to almost all browsing devices and accessible to almost all users.

- Create documents that work even if the user cannot see and/or hear by providing information that serves the same purpose or function as audio or video in ways suited to alternate sensory channels as well. This does not mean creating a prerecorded audio version of an entire site to make it accessible to users who are blind. Users who are blind can use screen reader technology to render all text information in a page.

It is important to remember that the content of a document refers to what it says to the user through natural language, images, sounds, movies, animations, etc. The structure of a document is how it is organized logically (e.g., by chapter, with an introduction and table of contents, etc.) The presentation of a document is how the document is rendered (e.g., as print, as a two-dimensional graphical presentation, as an text-only presentation, as synthesized speech, as braille, etc.) [11].

The WCAG document includes fourteen guidelines or general principles of accessible design. Each guideline includes a set of checkpoints that explain how the guideline applies to Web development. Each checkpoint is intended to be specific enough so that someone reviewing a page or site may verify that the checkpoint has been satisfied. Each of the checkpoints have an associated priority level assigned by the WCAG working group based on the checkpoint's impact on accessibility.

A separate document, entitled "Techniques for Web Content Accessibility Guidelines 1.0" [12], explains how to implement the checkpoints defined in the current document.

2.2.2 WCAG 2.0

Web Content Accessibility Guidelines (WCAG) 2.0 [9] covers a wide range of recommendations for making Web content more accessible. Following these guidelines will make the Web content accessible not only to a wider range of people with disabilities but will also often make it more usable to users in general.

WCAG 2.0 success criteria are written as testable statements that are not technology specific. Guidance about satisfying the success criteria in specific technologies, as well as general information about interpreting the success criteria, is provided in separate documents.

WCAG 2.0 succeeds WCAG 1.0. It is expected that according to W3C recommendation Web accessibility policies reference WCAG 2.0.

Although these guidelines cover a wide range of issues, they are not able to address the needs of people with all types, degrees, and combinations of disability. These guidelines also make Web content more usable by older individuals with changing abilities due to aging and often improve usability for users in general.

WCAG 2.0 foundation lies on four principles for Web accessibility perceivable, operable, understandable, and robust. Under each of this four principles there are 12 guidelines.

The 12 guidelines provide the basic goals that authors should work toward in order to make content more accessible to users with different disabilities. The guidelines are not testable, but provide the framework and overall objectives to help authors understand the success criteria and better implement the techniques.

For each guideline, testable success criteria are provided to allow WCAG 2.0 to be used where requirements and conformance testing are necessary such as in design specification, purchasing, regulation, and contractual agreements. In order to meet the needs of different groups and different situations, three levels of conformance are defined: A (lowest), AA, and AAA (highest).

For each of the guidelines and success criteria in the WCAG 2.0 document itself, the working group has also documented a wide variety of techniques. The techniques are informative and fall into two categories: those that are sufficient for meeting the success criteria and those that are advisory. The advisory techniques go beyond what is required by the individual success criteria and allow authors to better address the guidelines.

2.2.3 Section 508

The U.S. Access Board has issued access standards for federal electronic and information technology as required under Section 508 of the Rehabilitation Act: The Electronic and Information Technology Accessibility Standards.

The Access Board has published an online guide [6] for all the standards. This guide site is the easiest route to view the 16 provisions of the Section 508 Standards for the Web. The force of the Section 508 Standards is that electronic and information technology purchased by the U.S. federal government must comply with these provisions.

Because of that force of law, these provisions are seen as playing an important role in defining accessibility, especially in the U.S.

2.3 Mobile Web Content Standards

Mobile Web applications have been gaining momentum in the last years, due to the ever-increasing proliferation of Web-enabled mobile devices (especially smart phones).

It has been noted for several times that the constraints imposed by accessibility are akin to those imposed by the limitations of mobile devices. Examples such as properly structured information, correct (and linear) labeling of forms, or media equivalence of contents, are landmarks that illustrate this assertion. Consequently, striving for an accessible application is (partially) striving for a usable mobile application. Thus, a starting point to define a way to evaluate the accessibility of a mobile application is ensuring that in fact the application is usable in a mobile-centric environment.

The first problem in this scenario concerns the highly diverse ecosystem of mobile devices. There are different technology constraints imposed by devices' hardware fea-

tures (e.g. screen size, CPU, memory, connectivity), operating systems, APIs, UI Toolkits, look-and-feel, manufacturers' human-machine interface guidelines, programming languages, amongst others. For all of these reasons, it is very difficult to ensure that a mobile application is usable by any user in a holistic way (i.e., independent from any of these variables).

However, abstracting away from these constraints, there are general-purpose usability guidelines that can be applied to the mobile applications domain, as well as mobile-specific development guidelines that help building usable mobile applications.

2.3.1 Mobile Web Best Practices

W3C Mobile Web Best Practices document specifies best practices for delivering Web content to mobile devices [28]. The principal objective is to improve the user experience of the Web when accessed from such devices and the recommendations refer to delivered content and not to the processes by which it is created, nor to the devices or user agents to which it is delivered. It is primarily directed at creators, maintainers and operators of Web sites [28].

The document sets out a series of recommendations designed to improve the user experience of the Web on mobile devices. It refers primarily to the extension of Web browsing onto mobile devices [28].

The MWBP recommendations are in part derived from the Web Content Accessibility Guidelines (WCAG). WCAG guidelines are supplementary to the Mobile Web Best Practices, whose scope is limited to matters that have a specific mobile relevance [28].

The Mobile Web best practice statements address several requirements and issues of mobile usage namely presentation issues, input capabilities, bandwidth and cost, advertising.

Most Web pages are designed to be presented on desktop size displays, and exploit capabilities of desktop browsing software. Accessing such a Web page on a mobile device often results in a poor or unusable experience. Because of the limited screen size and the limited amount of material that is visible to the user, context and overview are usually lost for example. Frequently the subject matter of the page may require considerable scrolling to be visible, due to the limited size of the screen.

Mobile device input is often difficult when compared with use of a desktop device equipped with a keyboard. Mobile devices often have only a very limited keypad, with small keys, and there is frequently no pointing device. Because of the limitations of screen and input, forms are hard to fill in.

Mobile networks can be slow compared with fixed data connections and often have a measurably higher latency. This can lead to long retrieval times, especially for lengthy content. Furthermore Mobile data transfer often costs money.

Web pages can contain content that the user has not requested such as advertising or

large images. In the mobile world this contributes to poor usability and may increase the cost of the retrieval.

Some mechanisms commonly used for presentation of advertising material (such as pop-ups, pop-unders and large banners) do not work well on small devices and are therefore contrary to best practice recommendations.

Mobile browsers often do not support scripting or plug-ins and in many cases browser upgrading or change is not possible.

To cope with these and other specific mobile requirements, the Mobile Web Best Practices (MWBP) [28] guides how to make Web sites that are usable from a mobile device access. The objective is to improve the user experience of the Web when accessed from mobile devices.

MWBP define a set of checkpoints (akin to WCAG's) that developers must/should take into account, to ensure that a Web page or Web site is properly functional and tailored to mobile devices. MWBP checkpoints are aligned into 5 Best Practice Headings, as follows:

1. Overall Behaviour: general purpose guidelines for any mobile device, independent of its features;
2. Navigation and Links: how navigation and hyper linking should be done, in order to ease the task of interacting with Web-based user interfaces with the limited capabilities of mobile devices;
3. Page Layout and Content: how Web pages have to be designed, and how content must be created for mobile devices;
4. Page Definition: how to potentiate usability by exploiting the features of Web technologies;
5. User Input this is typically more restrictive on mobile devices than on desktop computers (and often a lot more restrictive). For example, mobile devices may lack pointing devices and often do not have a standard keyboard for text entry.

There is a subset of checkpoints that are machine verifiable, called mobileOK Basic Tests [26].

2.3.2 Mobile OK basic tests

The mobileOK Basic tests [26] are based on subset of the Mobile Web Best Practices [28]. Their outcome is machine-verifiable, hence claims of mobileOK Basic conformance can be checked.

The mobileOK Basic is a scheme for assessing whether Web resources (Web content) can be delivered in a manner that is conformed with Mobile Web Best Practices to a simple and largely hypothetical mobile user agent, the Default Delivery Context.

The Best Practices, and hence the tests, are not promoted as guidance for achieving the optimal user experience. The capabilities of many devices exceed those defined by the DDC. It will often be possible, and generally desirable, to provide an experience designed to take advantage of the extra capabilities.

The tests apply to a URI. Passing the tests means that when accessed through the rightly HTTP Request, resolving a URI will result in mobileOK Basic conformant content that is delivered in a mobileOK Basic conformant manner.

Individual tests may result in PASS or FAIL. PASS is required from all tests in order to claim mobileOK Basic conformance. In any test, PASS is achieved if and only if there are no FAILs. No specific PASS outcome is defined for any test.

Tests may also generate a number of informative warnings which do not affect whether a test has passed or not. A warning may indicate that it could not be conclusively determined whether the content under test conforms to a Best Practice (and thus does not FAIL), or may indicate that the content under test is close to violating a Best Practice.

Mobile OK Basic Tests comprehend a subset of MWBP namely for Auto refresh and Redirection, Chaching, character encoding, content format and valid markup, default input mode, external resources, graphics use for spacing, use of image maps, images resizing, images size specify, link target format, measures, frames existence, text-alternatives, objects or script existence, page size limit, page title existence, pop-ups, default existence, defaults provision, style sheets use and style sheets support, tables alternatives, tables layout and tables nested.

The intention of mobileOK is to help in the development of Web content that provides a functional user experience in a mobile context.

2.4 Assessment Tools

2.4.1 Accessibility assessment tools

Web accessibility assessment tools can be used to investigate the accessibility of a Web site and to implement accessibility features. There are several types of assessment tools which provide different features and characteristics [4].

Sometimes, a single tool may be adequately able to address the requirements of the developers but there are also circumstances where it may be suitable to select more than one tool. There are several types of tools which can assist in the development of accessible Web sites. These tools can generally provide one or more of the following features: 1. Evaluation: analysis of Web pages against a set of guidelines. 2. Repair: automated or semi-automated enhancement of the Web page markup to incorporate accessibility

features. 3. Transformation: modify the presentation of Web pages to assist Web users, but can also be used to identify potential barriers.

Web accessibility assessment tools are usually stand-alone applications but sometimes they can be plug-ins for authoring tools (such as editors, content management systems, or word processors), or Web browsers. A few assessment tools can also be configured to run on an ongoing basis to monitor the status of Web sites.

The Web Accessibility Initiative (WAI) presents a long list of Web accessibility evaluation tools [1]. These tools check the conformance with different sets of guidelines, analyze different kinds of content (HTML, PDF, etc.), or can be more specific, such as testing color contrast and brightness among other features.

Most widespread accessibility assessment tools include Sheriff Accessibility Module [18] from HiSoftware Compliance Tools, Wave [36] from WebAIM, or Lift Machine [34] from Usable Net allowing compliance for several guideline standards including section 508 and WCAG.

From the 114 assessment tools identified within the Accessible project for the accessibility assessment of Web pages and services, the majority use the WCAG1.0 guidelines either individually used or in combination with others [4].

The second most commonly used guidelines is the Sections 508 standards that are also used in many tools on their own or in combination, most usually with the WCAG1.0 guidelines. It is very important to notice that none of the tools used WCAG2.0 guidelines that are the most updated standards in the accessibility market right now. Thus ACCESSIBLE project intend to include them in its products design among with other guidelines to.

Because most of the assessment tools cover the majority of the existing accessibility guidelines, they can identify problems which are ultimately easy to fix, but often overlooked. It's often the case where there was no one to point them out. At the end of the process developers will see that by thoroughly going through the report and making minor or less minor changes to their Website, the degree of accessibility of their Website will have improved significantly.

However, these types of tools are focused on testing accessibility according to accessibility recommendations. Therefore, testing the accessibility of a Web page using those tools does not solve the problem of its adequacy to mobile devices.

2.4.2 Mobile adequacy assessment tools

The W3C provides a mobileOK checker [15] that follows the publication of the W3C mobile OK Basic Tests 1.0. It is a free service by W3C that helps check the level of mobile-friendliness of Web documents, and in particular assert whether a Web document is mobileOK. It has been designed as a tool to cope with the document that provide the basis for making a claim of W3C mobileOK Basic conformance, thus it does not go

further into accessibility assessment.

There are other evaluators such as the Ready.mobi from dotMobi [14] that uses industry standard tests developed with the W3C and leading mobility companies. This tool provides an analysis of how the Web content is likely to function on a mobile device. Many of the tests performed by ready.mobi are defined by the W3C in the MobileOK Basic Tests 1.0 document /citedotMobi:mobiReady01. This evaluator also supports simulation of the Web content visualization on some mobile devices.

T.A.W. OK basic [29] is another mobile Web checker based on W3C Mobile Web Best Practices 1.0. from CTIC technology center who also provides separate accessibility evaluation tools for WCAG 1.0 and WCAG 2.0. Mobile Web contents once again are tested in a certain extent for their adequacy to be accessed from mobile devices, but once again they are not evaluated regarding their accessibility to impaired users accessing those contents from mobile devices.

Other mobile devices testing tools provide answers to specific platforms testing, but again do not cope with accessibility issues or to Mobile Web Best Practices general guidelines.

2.5 Accessibility and mobile adequacy assessment

Regarding the way guidelines and best practices sets are viewed, several evaluation methodologies exist. Conformance testing evaluation methodologies assume that all accessibility guidelines must be met in order to achieve universal accessibility. Still, different groups of users have different requirements. Some of those requirements may conflict with each other and in many occasions for a specific user group the content of some guidelines does not constitute a barrier. Applying those would produce a false positive result that might lead a specific user group, erroneously, to avoid navigation through that page.

Barrier Walk through starts addressing this problem by providing a framework where guide-line application is related to specific user disability groups, such as blind users using screen readers, low vision users using screen magnifiers, motor-disabled users using a normal keyboard and/or mouse, deaf users, and cognitive disabled users [8]. An additional benefit from the method is the education of evaluators since they become more knowledgeable of accessibility issues with this approach than through the extensive and arid universal checklist evaluation using conformance testing [7].

An important extension to this work [38] defines Mobile Web Barriers, proposing mobile users as a group that has specific interaction limitations. Although an interesting approach, it fails to characterize the orthogonal nature of people and devices, which are clearly different conceptual and pragmatic entities. Moreover, in practice, precludes the introduction of the device dimensions and thus of its own specific characteristics (e.g. how to define barriers different barriers from a user with a small keyboard based device

vs. the same user with a touch based one).

Concerning the application of guidelines and best practices, different development phases and stakeholders can be targeted. Phases can be considered from the design and development to the final and intermediate evaluations. Stake-holders may vary from non-technical or expert evaluators, designers and developers. Either way though, the amount of information and intricacy that may arise from the several above mentioned dimensions, complemented by the demanding cognitive processes that are inherent to design, development and evaluation, urges for support that, as much as possible, automates the application of guidelines and best practices.

However, automation is not straightforward. In fact some of the regulations of WCAG and MWBP can not be automated. Fortunately though, a significant group is. For instance, the mobile OK basic tests recommendation [26] defines a set of tests based on MWBP to ease Mobile Web content authoring. It is a subset of MWBP with those best practices that can be pro-grammatically detected and/or verified in order to allow the development of concrete evaluation tools. A similar subset is defined to other recommendations. Even if the assessments based on those are not as complete, they will surely provide the pragmatic means for designers and developers to create less inaccessible and mobile non adequate Web contents. Moreover they can be easily complemented by manual evaluation of the remainder guidelines, or by formal or informal user participation, performed by significant sets of users from different dis-ability groups and skills.

For the automatable subsets, design and development platforms and tools exist that provide support to Web developers and designers to conduct rapid, yet specialized, accessibility assessments as referred in section 2.4.1 or compliance with the mobile Web best practices as referred in section 2.4.2.

As far as we know, no testing tools, services or methods provides the means to test Web content accessibility and mobile adequacy, considering the necessary flexibility and customization that we aim for. They either fail at coping with accessibility or with mobile access and mobile content adequacy, or specially ignore the specificities of disabilities in that context. To overcome this gap, there is a need for a new approach to evaluation processes, which we identify as Mobile Web Accessibility.

2.6 The ACCESSIBLE Project

The development of accessible software requires a strong effort from designers, developers and testers.

Taking into account different kinds of accessibility requirements, guidelines and best practices, and different user interface implementation technologies (which by themselves might pose severe problems of delivering accessible applications), developers are faced with a daunting task.

Therefore, the highly specialized skills required for developing accessible software sets aside most developers. To mitigate these problems, developers should be guided in their development process about accessibility concerns within software development.

This includes the definition of target users (e.g., their requirements, disabilities, etc.), which aspects should be taken into account to meet users' accessibility expectations, and how it reflects on software applications (thus coping with the particularities of different technologies).

To overcome the gap between developers knowledge on accessibility issues and the development of accessible and tailored software applications, ACCESSIBLE project will provide the potential users with an overall assessment and developer designer aid framework for the development of accessible software applications.

The overall layered architecture of the proposed system [33] is shown in Figure 2.1.

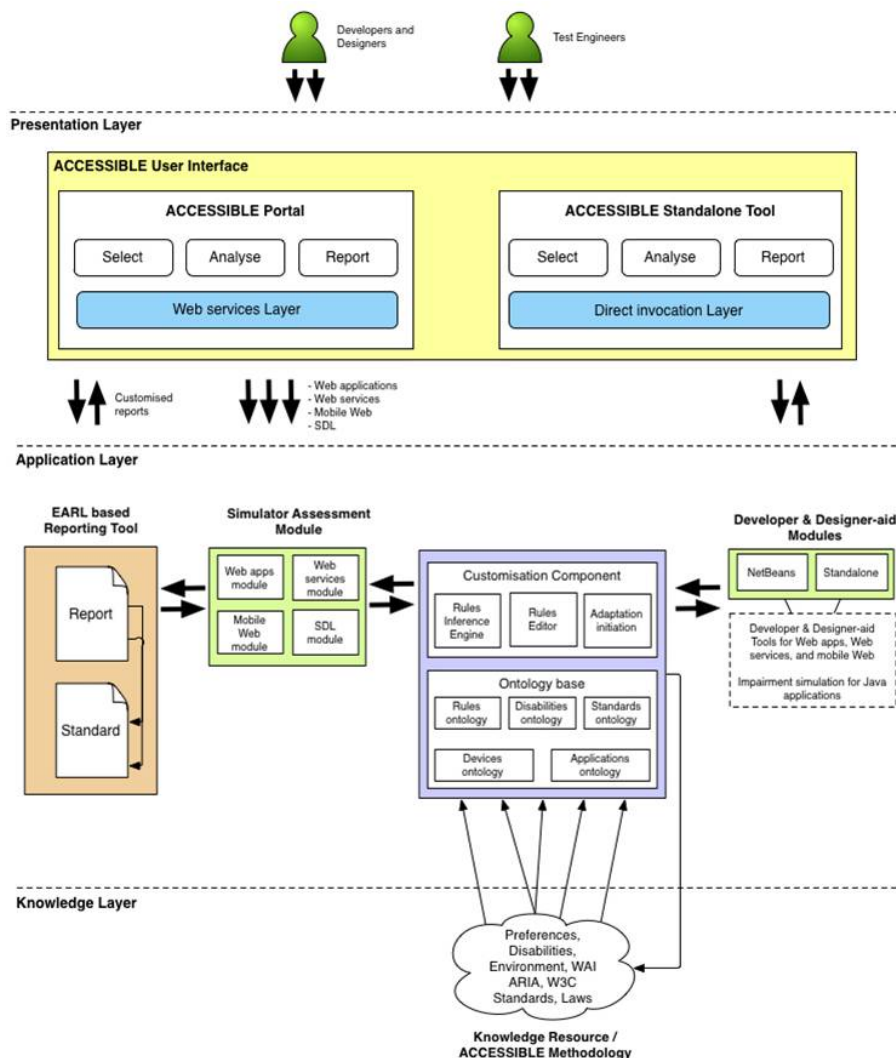


Figure 2.1: Overall Architecture of ACCESSIBLE System

The proposed architecture specifically addresses concerns about automation over ac-

cessibility testing and it compromises with independent modules that can interact each other. Towards this goal, the ACCESSIBLE working environment develops a modular framework that provides the interface between users and the following interacting components:

- ACCESSIBLE Portal/ User Interface of the system
- The assessment simulation modules
- The ACCESSIBLE ontological Knowledge Resource
- The developer/designer-aid module
- The generated analysis reporting tool.

Different architectural approaches are relevant for the ACCESSIBLE user interface architecture as identified below:

- An overall Web user interface portal that can support users that they would prefer to verify the accessibility of their applications online with the adoption of relevant Web services.
- Different standalone user interfaces for the open source standalone applications that can be downloaded to users terminals and can be used for the accessibility assessment of their software components

In terms of appearance, the ACCESSIBLE Web user interface can be thought of as a portal where relevant users can use the Web based services of the ACCESSIBLE components, to extract useful information for accessibility guidelines, standards, etc. and finally to download the standalone modules of the ACCESSIBLE project. That access will take place using a common Internet browser.

The portal interacts with all the main components of the ACCESSIBLE system such as the assessment simulation module, the ontology, the designer aid module and the EARL reporting tool.

The Assessment Simulation module will support the overall analysis and verification in terms of accessibility for Web applications, Web services, Mobile applications and Description Languages (UML, SDL, etc.). The module, which takes input from the ACCESSIBLE knowledge resource, will be composed of independent accessibility assessment tools in order to support the overall accessibility assessment process. This module includes:

- A Web applications assessment tool (Web and Standalone version) for the accessibility verification of Web applications.

- A Web services assessment tool for the accessibility verification of Web services
- A Mobile Web Content Assessment Module for the accessibility verification of mobile Web contents
- An Description Language Assessment tool

The mobile Web content assessment tool proof of concept is within the focus of this work.

2.6.1 The Harmonized Accessible Methodology

As mentioned before the rules applied depend on the disability group profile. As part of the ACCESSIBLE project, Accessibility Assessment Simulation Environment for New Applications Design and Development, an Accessible Harmonized Methodology (HAM) is being developed.

The purpose of HAM [10] is the harmonization of existing knowledge, such as guidelines, standards, etc. in order to be described by ontology-based rules. The resulting framework will allow the implementation of automated assessment systems, enabling designers, programmers, evaluators, disability group users, etc., to conduct specialized accessibility assessments focused on specific disability types, assistive technologies, platforms and/or contextual conditions.

The International Classification of Functioning, Disability and Health, commonly known as ICF [25], are at the core of HAM. ICF provides a concrete classification of body structure impairments which ensures no overlapping. This can be linked to user types (e.g., disability types) in order to link them to ICF body structures and their related impairments. Body structure impairments which are deviation or loss of body functions such as sight, hear, etc., are mapped to the resulting interaction limitations such as vision loss creates a blind spot, blur.

Those interaction limitations are associated with guidelines and heuristics which allows to determine which user groups, such as different disability type groups or subgroups benefit from each guideline application.

Our particular work focused on the mobile extension to cope not only with the associated WCAG guidelines but also with the mobile Web best practices guidelines and tests. The HAM approach can be seen in figure 2.2

This mapping of ICF body structure impairments into interaction limitations facilitates the association of existing guidelines and heuristics from the existing literature to specific body structures and therefore to disability user groups, allowing determining which users groups would benefit from each guideline application. Figure 2.3 shows an example this mapping.

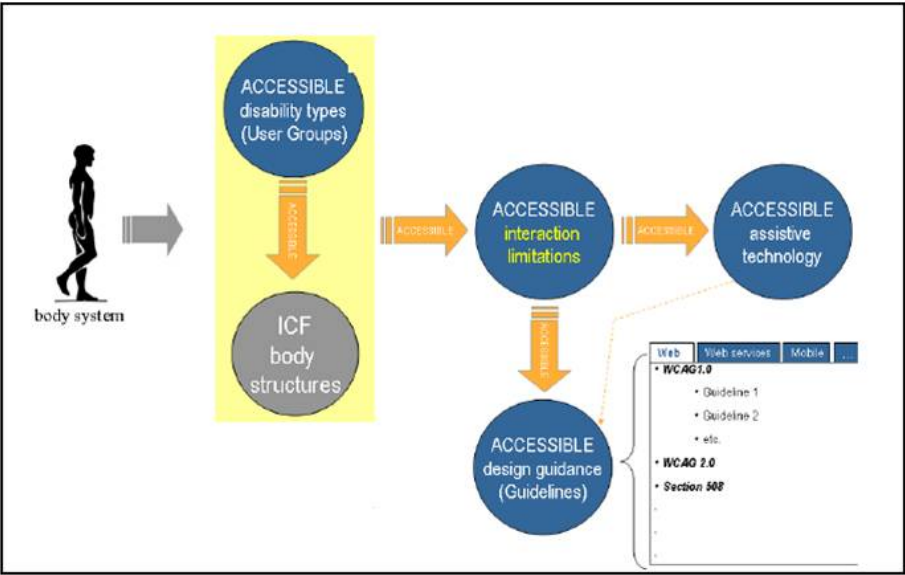


Figure 2.2: HAM Approach

Disability Type		Interaction limitation	
Main Type	Subtype	Function	Checkpoints
Cognitive impairment	Dementia	May have difficulties in language, self-help, independent living etc. They acquire new knowledge at a slower pace than their peers. They have difficulties in understanding instructions. There may be difficulties in fulfilling the every-day duties or organizing one's own work load. These may be difficulty in remembering people's names, a telephone number or important address. In case there are problems in retrieving memories, there may be difficulties in a number of	<ul style="list-style-type: none">○ Generic links○ Rich images lacking equivalent text○ Video with no captions○ Moving content○ Generic links○ Internal links are missing○ Missing icons○ Complex of text○ Complex of site○ Too many links

Figure 2.3: Disability type associated Interaction Limitations example

The approach of HAM regarding Mobile Web Contents is focused on the dichotomy between the constraints imposed by accessibility and mobile domains. Thus, a primary reflection will be made on how to extend the HAM work on mapping Disability Types to WCAG 1.0, WCAG 2.0 and Section 508 guidelines into the mobile constraints emerging from the MWBP.

The mapping of Web content guidelines, WCAG 1.0, WCAG 2.0 and Section 508 to disability types and interaction limitations result from the design guidance work for Web content performed in HAM.

These guideline correlation between accessibility guidelines and its associated assessment rules and tests are extremely relevant to our work since they have to be also considered when evaluating the mobile usage adequacy of Web contents for disabled groups of users.

Figure 2.4 shows an example this association.

Disability Type		Short Description in terms	ICF classification	Interaction limitation		Assistive Tech.	Guideline		
Main Type	Subtype			Function	Checkpoints		WCAG 1.0	WCAG 2.0	Section 508
Vision impairments	Low vision • Loss of central vision (Macular degeneration)	Macular degeneration is a medical condition usually of older adults which results in a loss of vision in the centre of the visual field (the macula) because of damage to the retina.	b210 Seeing functions b2100 Visual acuity functions b21000 Binocular acuity of distant vision b21001 Monocular acuity of distant vision b21002 Binocular acuity of near vision b21003 Monocular acuity of near vision b21008 Visual acuity functions, others unspecified	The loss of central vision creates a blur or blind spot, but side (peripheral) vision remains intact. This makes it difficult to read, recognise faces, and distinguish most details in the distance. Mobility, however, is usually unaffected because side vision remains intact.	<ul style="list-style-type: none"> Rich images that are badly positioned Rich images included in the background Colour is necessary Insufficient visual contrast Inaccessible frames Moving content 	Refreshable Braille Display Braille Embosser Portable Closed Circuit TV Reading Machine Closed Circuit TV	1.1	1.1	1194.22 (a)
							2.1, 2.2	1.1.1	1194.22 (c)
							3.4	1.3	1194.22 (d)
							6.1, 6.3	1.3.1, 1.3.2	1194.22 (f)
							7.5	1.4	1194.22 (m)
							8.1	1.4.1, 1.4.3	1194.22 (e)
							12.1, 12.2	1.4.4, 1.4.6	
							10.1	2.2	
							13.6, 13.8	2.2.2	
								2.4	
								2.4.1	
								3.2	
								3.2.5	
								4.1.1, 4.1.3	

Figure 2.4: Example of best practice catalogue for Web Applications

To the above mentioned mapping of WCAG 1.0 and WCAG 2.0 guidelines to disability types we extended the set of guidelines to include the MWBP set as we will present in the next chapter.

2.7 Summary

This section presented this thesis related work. First disabled mobile use and barriers are introduced. Then Web content accessibility main standards are presented and described. Afterwards, Mobile Web contents standards are also presented. Then the existing accessibility assessment tools and the mobile adequacy tools are discussed as well as their existing gaps. This chapter ends presenting the ACCESSIBLE project within which scope this thesis work in undertaken and its accessibility assessment methodology.

Chapter 3

Analysis

This chapter discusses the introduction of the mobile dimension on the accessibility evaluation process where different disability profiles are considered. The identification of the exact guidelines subset that should be applied on each profile is analyzed, having in consideration the MWBP and WCAG standards and the approach put down by HAM. Next, the requirements specification for the MWAAT (Mobile Web Accessibility Assessment Tool) is presented, including the identification of the target users groups and use cases' diagrams and description. Classes are briefly described as well as non-functional requirements.

3.1 Mobile Web Accessibility for Disability Profiles

Accessible mobile Web contents imply taking into consideration the hardware and technological constraints of mobile devices usage as well as how these impose constraints to the different types of disabilities. In general thus, the objective is to find a coherent subset of the combination of MWBP and WCAG guidelines that should be applied to assess some Web content for a specific disability type when using a mobile device.

3.1.1 WCAG checkpoints on a mobile usage setting

Considering only WCAG checkpoints, its relevance is clearly dependent on the disability type, as was described in the previous chapter and consolidated by the HAM. The table 3.1 shows an exemplifying and representative subset of such dependence.

Analyzing the WCAG guidelines and checkpoints, and considering the context when they apply, one reaches the conclusion that the accessibility checkpoints that are relevant to a disability type when using a desktop does not change on a mobile setting. Instead they tend to be reinforced by the aforementioned relation between WCAG and WMBP. Again, the checkpoints denoted in the table illustrates this observation.

Checkpoints	Disability Type					
	Blind	Low-Vision	Deaf	Color-Blind	Motor Impaired	Cognitive
checkpoint 1.1	x	x	x		x	x
checkpoint 1.4	x		x			
checkpoint 2.1	x	x		x		
checkpoint 2.2		x		x		
checkpoint 6.3	x	x			x	
checkpoint 6.4	x	x			x	
checkpoint 9.3	x				x	

Table 3.1: Example of relevant checkpoints for disability type - a cell with x means that the checkpoint is relevant for that disability type

3.1.2 MWBP for disability types

Regarding MWBP, three subsets of guidelines should be considered [3], namely:

- Guidelines directly relevant to one or more disability types;
- Guidelines relevant to mobile device usage regardless of the users' special needs;
- Guidelines that become irrelevant for some disability types when accessing from mobile devices.

To reach the first subset, one should depart from the accepted relation between the MWBP and WCAG. The correspondence between MWBP guidelines, interaction limitations and disability types can be straightforwardly derived. In fact, assuming the mapping established on HAM from disability type to interaction limitations, and the above mentioned relation, we will obtain a subset of MWBP tests. The figure 3.1 shows that mapping for the Visual Impairments disability type.

Disability Type		Short Description in terms	ICF classification	Interaction limitation		Assistive Tech.	Guideline MWBP
Main Type	Subtype			Function	Checkpoints		
Vision impairments	Low vision • Loss of central vision (Macular degeneration)	Macular degeneration is a medical condition, usually of older adults which results in a loss of vision in the centre of the visual field (the macula) because of damage to the retina.	b210 Seeing functions b2100 Visual acuity functions b21000 Binocular acuity of distant vision b21001 Monocular acuity of distant vision b21002 Binocular acuity of near vision b21003 Monocular acuity of near vision b21008 Visual acuity functions, others specified b21008 Visual acuity functions, unspecified b2101 Quality of vision b21023 Visual picture	The loss of central vision creates a blur or blind spot, but side (peripheral) vision remains intact. This makes it difficult to read, recognise faces, and distinguish most details in the distance. Mobility, however, is usually unaffected because side vision remains intact. Individuals with Macular Degeneration have extreme difficulty reading and recognising faces. Shapes or contours of objects may seem distorted, yet enough peripheral vision is retained for good orientation and mobility.	<ul style="list-style-type: none"> o Rich images that are badly positioned o Rich images included in the background o Colour is necessary o Insufficient visual contrast o Inaccessible frames o Moving content o Image maps o Too long toolkits o Internal links are 	Refreshable Braille Display Braille Embosser Portable Closed Circuit TV Reading Machine Closed Circuit TV	ACCESS KEYS AUTO REFRESH BACKGROUND_IMAGE_READABILITY CLARITY COLOR_CONTRAST FONTS GRAPHICS_FOR_SPACING MEASURES MWBP NON-TEXT_ALTERNATIVES OBJECTS_OR_SCRIPT PAGE_SIZE_USABLE POP_UPS REDIRECTION

Figure 3.1: Mapping Disability Type to MWBP

The meaning of this subset, though, should be carefully assessed. First, the relation of these guidelines, WCAG and MWBP, do not always correspond to a full equivalence or even an implication. That means that, most of the times, both must be checked - either in terms of failing/warning condition or/and in terms of reporting situations.

Compliance with WCAG 1.0 helps towards achieving compliance with some of the MWBP, and compliance with the MWBP helps towards achieving compliance with some WCAG 1.0 checkpoints. Many MWBP have the added benefit of partial or complete compliance with certain WCAG success criteria or checkpoints. However it should not be assumed that following any MWBP will ensure accessibility or that a success criterion or checkpoint will ensure compliance with MWBPs. To ensure compliance it is important to always consult the Mobile Web Best Practices or the Web Content Accessibility Guidelines directly [13].

For example WCAG checkpoint 5.3 “Do not use tables for layout unless the table makes sense when linearized (Priority 2)”, is partially covered by TABLES_LAYOUT that states “Do not use tables for layout”, TABLES_SUPPORT that states “Do not use tables unless the device is known to support them” and TABLES_ALTERNATIVES that states “Where possible, use an alternative to tabular presentation”. Although the three mentioned MWBP state recommendations restricting table usage derived from the fact that on mobile devices limited size screens tables do not work well and may result in the user having to scroll horizontally and vertically, none of them tests the significance of the content when linearized, an issue relevant for a blind or low-vision user using an assistive technology with screen reader capabilities. And although TABLES_ALTERNATIVES is possibly partially covered at priority 2 by 5.3, it is not completely covered [13].

Secondly, the mapping does not exclude all the other MWBP guidelines just because there is a relation between some parts. There are also aspects important for mobile usage adequacy that do not relate to accessibility specific issues or to WCAG checkpoints. Guidelines regarding features such as character encoding, clarity, content format preferred, content format support, cookies, etc. do not specifically have a relation with a specific WCAG best practice issue, but rather are critical to general mobile devices interaction. The PAGE_SIZE_LIMIT or the LINK_TARGET_FORMAT guidelines are good examples of these. Generally, they must (as others) be satisfied by all mobile Web content and thus by the application designers in order to create an accessible Web content that is adequate to be used on mobile devices.

In principle, then, the conjunction of these two MWBP subsets, i.e., WCAG related and accessibility independent, along with the disability specific WCAG set will constitute the whole relevant set that should be used to assess content for given disability. However, a deeper analysis revealed some interesting, potentially controversial, issues.

Consider a blind disability type or a user that by rule turns off the images download option on its user agent of his/her mobile device. Applying MWBP image related tests for guideline conformance (e.g. images size specification) can result in failure results irrelevant for that specific usage. In fact, not having the image size specification will not change at all the user experience since the image will not be downloaded anyway.

Like the above example, others can arguably be pointed, that if taken into consideration will avoid false positive test results. Thus, a tool or a methodology that supports disability profiles in a coherent mobile Web accessibility evaluation, must definitely have to reference these false positive guidelines. At the best, they should be treated as particular cases, probably of low level severity, if not simply removed from the specific disability relevant guideline set.

The table 3.2 shows a set of WMBP guidelines that may be considered irrelevant (those not marked with an “x”) for assess mobile Web content directed to a particular disability type.

Guideline	Disability Type					
	Blind	Low-Vision	Deaf	Color-Blind	Motor Impaired	Cognitive
BACKGROUND_IMAGE_READABILITY			x	x	x	x
CONTROL_POSTION			x	x	x	x
FONTS			x	x	x	x
IMAGES_RESIZING			x	x	x	x
IMAGES_SPECIFY_SIZE			x	x	x	x
LARGE_GRAPHICS			x	x	x	x
MEASURES			x	x	x	x
SCROLLING			x	x	x	x
STYLE_SHEETS_SIZE			x	x	x	x
STYLE_SHEETS_SUPPORT			x	x	x	x
STYLE_SHEETS_USE			x	x	x	x

Table 3.2: Relevant MWBP guidelines for disability type - a cell with x means that the checkpoint is relevant for that disability type; an empty cell means that relevance is minimal or absent

Considering the low-vision disability type, some aspects are noteworthy to mention. Some low-vision users can read some information on desktop monitors using operating system or browser’s magnifying capabilities. Others won’t be able, even on large desktop displays, to access Web content without using an assistive tool with some screen reading capabilities. On existing mobile device displays low-vision impairment is further stressed by the small display size. Consequently on table 3.2 we consider that low-vision users do not have the capabilities to access mobile Web contents without an assistive technology use such as a screen reader.

3.2 Requirements Specification

The target user groups of MWAAT (Mobile Web Accessibility Assessment Tool) are the professional users, such as developers and designers, but also considering anyone that wishes to test the accessibility of pages and its mobile adequacy. We thus will consider the following two actors:

- **Testers** which can be disabled users, disabled user group representatives or assessment experts wanting to check Web resources regarding its adequacy to mobile devices and disability profiles.

- **Developers** are users directly involved in the design, development and test of Web resources. It will normally involve an iterative process as these users will need to recurrently evaluate their own work also during its development. Thus, at this level, MWAAT could easily play the role of a debugger, for mobile and accessibility evaluation purposes. Developers are expected to use MWAAT directly on their own source code files even before they are deployed on the destination Web servers, and of course, afterwards.

Testers' use cases are a subset of the use cases required by a developer. For simplification reasons only the latter is considered henceforward.

Besides the two human-based actors, the MWAAT system interacts also with a third supporting actor, the HTTP Server, which must be taken into account. This will be responsible for handling the HTTP requests issued by the MWAAT as well as handling the HTTP responses received from the Web Servers.

3.2.1 Use Cases

Figure 3.2 shows the use case context diagram for the MWAAT. Each use case is briefly presented in the current section. The presentation will follow a Developer actor centered approach, from higher level cases to detailed ones in a dept first approach.

- **Assess Web Resource:** The user needs to be able to select the Web Resource and define the Usage Scenario he/she wants to evaluate. The latter is determined by the Select Relevant Guidelines use case; the former by conjunction of the following three use cases. The Web resource must be requested and obtained through the HTTP Server actor.
- **Choose URL:** In order to do the assessment of a Web content the user needs to be able to set the Uniform Resource Identifier (URI) of what he wants to evaluate.
- **Select Delivery Context:** The user needs to be able to select the delivery context simulating the access of its choice. Different representations of the same URI can be provided by the Web server depending on the device or user agent that is accessing it. For example, a Web page representation might be different whether the access is made from Chrome, Firefox or IE browser, or more importantly in the case of this work, if it is done from a desktop or a mobile device.
- **Set Proxy:** In some operation environments access to the Web is performed through a proxy server. The user needs to be able to set the Proxy Server.
- **Set Relevant Guidelines:** The user needs to be able to select the relevant guidelines and corresponding tests that will be used in the Web content assessment. This will be achieved selecting: the disability type and the device type.

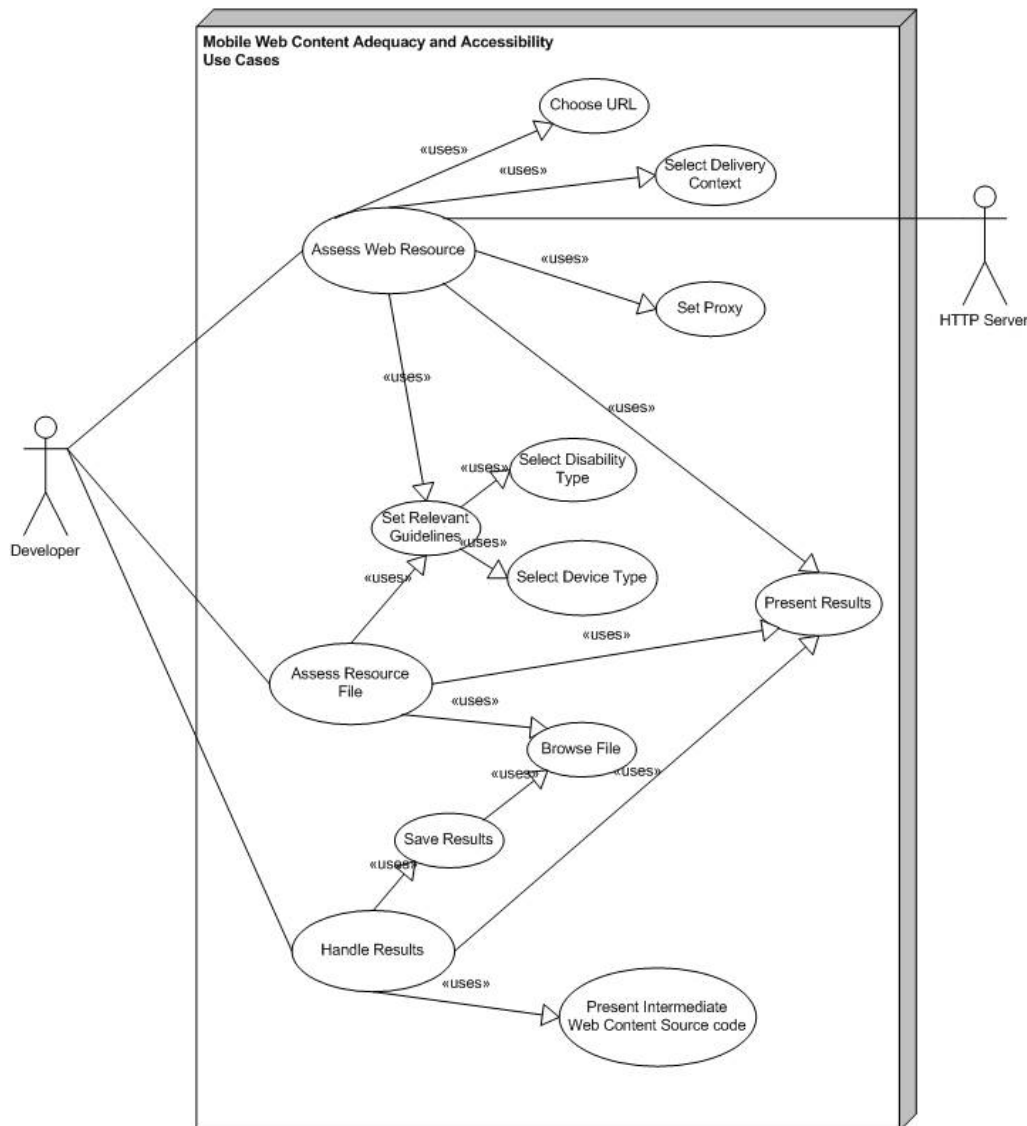


Figure 3.2: Use Case diagram for MWAAAT

- Select Disability Type:** The user needs to be able to select one of the available disability types. Each corresponds to different impairments, resulting in different interaction limitations. Thus, the disability type choice will determine the set of relevant accessibility guidelines and corresponding tests that will be used in the assessment, in accordance to the proposed methodology.
- Select Device Type:** The user needs to be able to select the target device to which the Web content will be delivered. This choice will result in different capabilities (e.g., display size, input capabilities, etc.) to reproduce the selected Web resource. Thus, the device type choice will determine the set of adequacy tests that will be used in the assessment, in accordance to the proposed methodology. Moreover, the combination of Disability and Device Type's selection will further refine the choice

as referred previously.

- **Assess Resource File:** Developers need to assess the Web Content of an HTML file that is not yet deployed on a Web server.
- **Browse File:** Developers need to browse the file system to search/select the file or directory they want to access.
- **Handle Results:** The user needs to review the assessment results, frame them in the actually evaluated Web source representation and keep them for later analysis.
- **Present Results:** The system will present the evaluation results to the user in a manner that is compliant with the standards and the user needs.
- **Present HTML Source Code:** The system will present the evaluated HTML source code. This coincides with the selected file for a resource file assessment or to the representation returned by the HTTP Server when a Web resource assessment was issued.
- **Save Results:** The user needs to be able to save the assessment results. Usually he/she will use the Browse File use case to define the filename and directory where the results are saved.

The Assess Resource File is only relevant to users that need to assess under development Web pages not yet uploaded to a Web server, namely developers. Testers, as well as end users or public and governmental bodies users, will generally have access to the remainder subset of the developers use cases.

3.2.2 Use Cases Definition

In this section we present the MWAAT use cases structured description comprising several sections whose meaning are:

- **Summary:** brief description of the use case.
- **Rationale:** the reason why the use case is needed.
- **Actors:** list of user categories that interact with this use case.
- **Preconditions:** the state of the software when the use case begins
- **Basic Course of Events:** list of interactions between the system and the involved users.
- **Alternative Course of Events:** conditions under which the basic course of events could change.

- **Post Conditions:** the state of the software when the use case ends.

Name	Assess Web Resource
Summary	The user wants to evaluate a Web Resource content accessibility and mobile adequacy according to the settled usage scenario
Rationale	The user needs to see what characteristics of the source code might raise barriers to the effective accessibility of the selected Web content.
Actors	Developer, Tester
Preconditions	The user is in the initial screen
Basic Course of Events	<ol style="list-style-type: none"> 1. The user selects the assessing option 2. The system performs the HTTP request 3. The system handles the HTTP status and response 4. The system parses the HTTP response content 5 The system applies the current active list of tests to the HTTP response contents and HTML document elements 6 The system presents to the user the assessment results
Alternative Course of Events	<ol style="list-style-type: none"> 1. The user selects the assessing option 2. The system performs the HTTP request 3. The system handles an HTTP unsuccessful status 4 The system presents the error messages to the user
Post Conditions	The assessment results and intermediate source code were obtained

Name	Choose URL
Summary	The user inputs the Uniform Resource Locator of the Web content he wants to access in order to perform an assessment
Rationale	In order to assess the accessibility and mobile adequacy of a given Web content available on the Web the user needs to be able to input the Uniform Resource Locator to access that content
Actors	Developer, Tester
Preconditions	The user is in the initial screen
Basic Course of Events	1. The user inputs the Uniform Resource Locator
Alternative Course of Events	None
Post Conditions	The URL is settled.

Name	Select Delivery Context
Summary	The user might select to simulate a mobile OK access to the Web resource or to simulate a regular default desktop access to the Web resource
Rationale	Web servers might send alternate resources representations to different delivery contexts namely the default representation and the mobile representation based on the HTTP headers headers. Different representations might have different evaluation results when evaluated regarding mobile accessibility.
Actors	Developer, Tester
Preconditions	The actor is on the application main or initial screen.
Basic Course of Events	<ol style="list-style-type: none"> 1. User selects the desired delivery context from the ones available 2. System sets the HTTP request parameters in order to simulate the pretended request
Alternative Course of Events	None
Post Conditions	The current simulation scenario is settled according to the chosen simulation and the corresponding request parameters are settled accordingly.

Name	Set Proxy
Summary	Setting the proxy sever and the associated port number
Rationale	Some networks require Internet access through a proxy server. The user needs to be able to set the proxy and port number. Since the proxy server and port number remain usually unchanged, the user might want to save its setting in order to be able to reuse them.
Actors	Developer, Tester
Preconditions	<ol style="list-style-type: none"> 1. The user selected the configuration option 2. The set proxy screen is active
Basic Course of Events	<ol style="list-style-type: none"> 1.The user enters the proxy server name 2. The user enters the port number 3. The user saves its input. 4 The user sets the proxy server and port number.
Alternative Course of Events	Alternate Course 1: <ol style="list-style-type: none"> 1.The user loads the proxy server name and port number 2. The user sets the proxy server and port number. Alternate Course 2: <ol style="list-style-type: none"> 1.The user resets the proxy server name and port number
Post Conditions	The current proxy server and port number are settled. Any HTTP requests will be performed through the settled proxy server.

Name	Set Relevant Guidelines
Summary	The user sets the relevant guidelines to be considered in the assessment
Rationale	The disability type corresponding to different interaction limitations is mapped to the relevant guidelines and associated tests that should be taken into account when assessing the Web content for that specific disability user group. According to the methodology for each disability type the set of tests are different according to the type of device used to access the Web content. If a mobile device is used, the relevant mobile Web best practices derived tests must be also applied.
Actors	Developer, Tester
Preconditions	The actor is on the application main or initial screen.
Basic Course of Events	1. The user selects the disability type 2. The user selects the device type
Alternative Course of Events	1. The user selects the device type 2. The user selects the disability type
Post Conditions	The relevant guidelines are selected and presented

Name	Select Disability Type
Summary	The User selects a disability type from the predefined set of disability groups available
Rationale	To assess the Web content with the appropriate tests, the user needs to select the disability type. The disability type corresponding to different interaction limitations is mapped through the methodology to the relevant guidelines and associated tests that should be taken into account when assessing the Web content for that specific disability user group.
Actors	Developer, Tester
Preconditions	The user is on the application main or initial screen.
Basic Course of Events	1. User Selects a Disability Type 2. System evaluates the tests that correspond to this disability type and takes into consideration also the current device type selected 3. System presents the corresponding test list to the User
Alternative Course of Events	None
Post Conditions	The current active list of tests that will be used to assess the Web resources are presented to the user. They are dependent on the currently selected device and the disability type just selected.

Name	Select Device Type
Summary	The User selects a device type from the predefined set of device types available
Rationale	Different devices have different characteristics and derived limitations. These characteristics introduce design and development issues that must be assessed through specific tests derived from mobile Web content standards and guidelines. Initially two major scenarios should be taken into account, the default access and the mobile access.
Actors	Developer, Tester
Preconditions	The actor is on the application main or initial screen.
Basic Course of Events	<ol style="list-style-type: none"> 1. User Selects a Device Type 2. System evaluates the tests that correspond to that device type and the current disability type chosen 3. System presents the corresponding tests list to the User
Alternative Course of Events	None
Post Conditions	The current active list of tests that will be applied to the Web resources of the current test list derived from the current disability type and the device type just selected.

Name	Assess Resource File
Summary	User choses to evaluate a Web Content source file
Rationale	The user, a developer need to evaluate the source code he is producing even before the site is installed on a Web Server
Actors	Developer
Preconditions	The user is on the HTML file evaluation screen
Basic Course of Events	<ol style="list-style-type: none"> 1. The user selects the file evaluation option 2. The system reads the file. 3. The system parses the HTML document obtained from the file 4. The system applies the current active list of tests to each of the HTML document nodes 5. The system presents the evaluation results to the user
Alternative Course of Events	
Post Conditions	The system sets the corresponding assessment results and analyzed source code

Name	Browse File
Summary	The user browses the system files in order to select one
Rationale	In order to assess the content of a source file under development or in order to choose the file where he wants to save the evaluation results, the user needs to browse the file system until he finds the wanted file path
Actors	Developer
Preconditions	The user is on the file browser screen
Basic Course of Events	<ol style="list-style-type: none"> 1. The user chooses to select a file 2. The file screen browser is presented to the user 3. The users selects the file
Alternative Course of Events	<ol style="list-style-type: none"> 1. The user chooses to select a file 2. The user cancels the file choice
Post Conditions	The filename is settled to the selected filename or empty

Name	Handle Results
Summary	The user chooses to see or save the evaluation results or to see the source code of the Web content that was parsed and analyzed.
Rationale	The user needs to have access to the results of the evaluation performed, needs to have the possibility of saving the results for later analysis or work and needs to have access to the HTML source code of the Web content evaluated.
Actors	Developer, Tester
Preconditions	The user is on the initial screen
Basic Course of Events	<ol style="list-style-type: none"> 1. The user selects the handle results option available in the system 2. The user selects the option to present the results on the screen or to save the results in a file or to present the source code on the screen.
Alternative Course of Events	None
Post Conditions	The system presents the options to see the last source code evaluated or to see the results obtained in the last evaluation

Name	Present Results
Summary	The user chooses to see the results of the last assessment
Rationale	The user wants to go through and analyze the results obtained on the last assessment
Actors	Developer, Tester
Preconditions	The user is on the handle results menu
Basic Course of Events	1.The user selects the option corresponding to the results presentation 2 The system clears any other information that was on the output area of the screen 3 The system lists the results on the presentation area 4 The system lists allows the user to navigate through the results of the last assessment
Alternative Course of Events	None
Post Conditions	The content of the presentation area shows the results of the last assessment

Name	Present HTML Source Code
Summary	User selects to see the source code of the Web resource he chose to evaluate
Rationale	In order to understand the results of the evaluation it is important to the developer to see the source code of the Web resource representation under evaluation.
Actors	Developer, Tester
Preconditions	The user is on the handle results menu
Basic Course of Events	1. The user selects a menu option to see the source code
Alternative Course of Events	None
Post Conditions	The tool output area shows the source code corresponding to the HTTP response content that was assessed.

Name	Save Results
Summary	User selects a file to save the results obtained from the last assessment
Rationale	Web contents change over time and the access to a URL implies the access to the Web. For those reasons or even others, the user might want to save the results he obtained
Actors	Developer, Tester
Preconditions	The user is on the initial screen
Basic Course of Events	<ol style="list-style-type: none"> 1. The user selects the save results option 2. The user chooses a file to save the results 3. The system saves the last results in the chosen file.
Alternative Course of Events	<ol style="list-style-type: none"> 1. The user selects the save results option 2. The user chooses to save the results to the current settled save results file
Post Conditions	The chosen file contains the last results obtained

3.3 Class diagrams

In this section the initial domain model is presented in order to identify the conceptual classes related to the current requirements. The following main system areas emerge from the previous use cases and requirements:

- Handling the resource representation that will be evaluated;
- Selecting the evaluation conditions, meaning the set of relevant guidelines and their inherent assessment tests, that derive from the high level user choices (disability and device types) and materialize the proposed methodology;
- Perform the evaluation, i.e., execute each test on the source representation gathering the evaluation results;
- Present the results to the user.

The class diagram corresponding to the system is represented in the figure 3.3. The figure materializes the four main areas in classes Resource-Handler, Selector, Evaluator and Results-Handler, respectively. The four classes are linked to the MWAAAT class that manages the whole set.

The Resource-Handler class manages the options that the user has for resource specification. The simpler version, HTML-file class, can access directly to the file resource that the user aims to evaluate. The more elaborated one, URL class, should consider the delivery context, meaning the user agent simulation, and the user specified URI in order to issue, to the HTTP Server actor, the required HTTP request, regarding the delivery context characteristics, as well as handling the associated HTTP response. The upper right side of the diagram illustrates this articulation

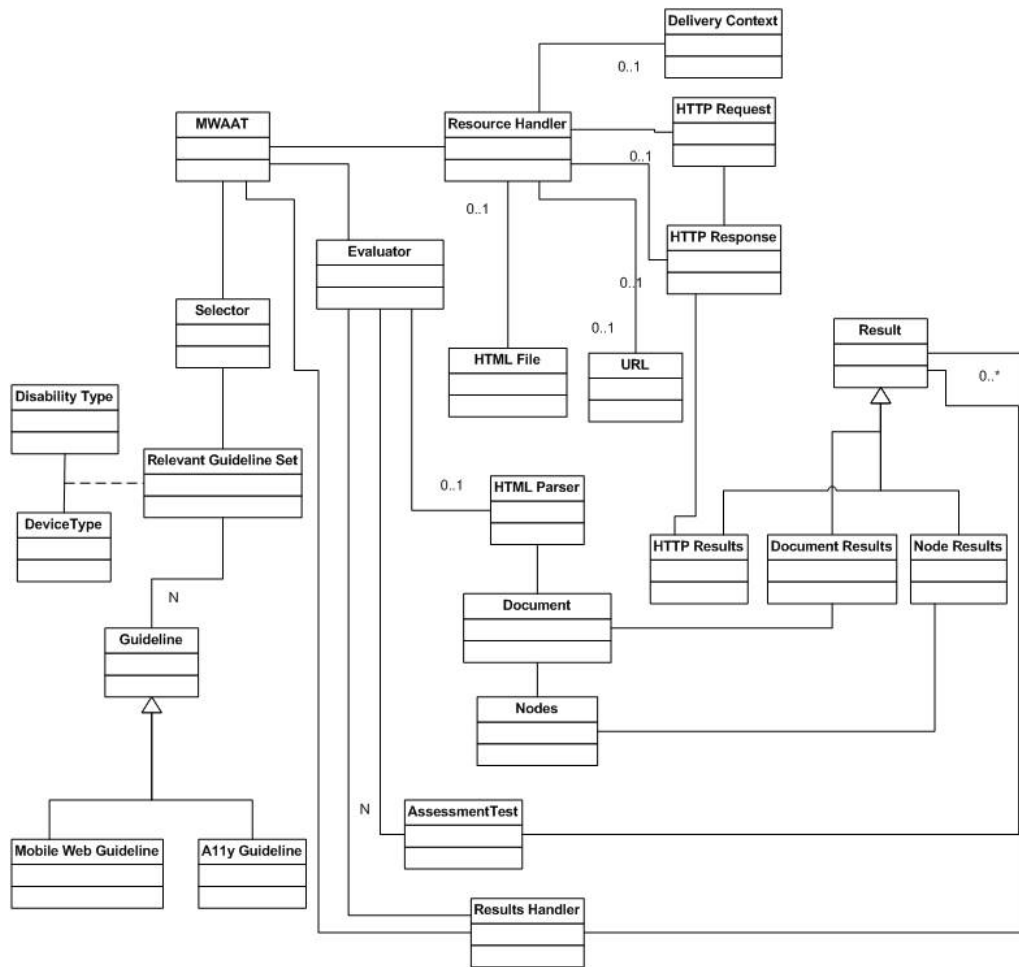


Figure 3.3: MWAAT Main Class Diagram

The Selector class manages the mapping between Disability and Device Types combinations and the corresponding assessment Guidelines. Each of those different combinations will have its own associated Relevant Guidelines Set, a selection of the whole guidelines' set according to the proposed methodology. Guidelines, of course, are either Mobile Web Guidelines derived from the Mobile Web Best Practices or Accessibility Guidelines derived from the Web Content Accessibility Guidelines.

The Evaluator class is responsible for applying the Assessment Tests to the resource handled by the Resource Handler. The assessment tests are determined by the guidelines set obtained by the Selector. Primarily, the Evaluator is accountable for parsing, HTML Parser class, and obtaining the document tree of Nodes from the HTML Document. Then, for each node the Evaluator will perform each of the relevant assessment tests.

It is worth notice that, in order to support extensibility of the set of available assessment tests, either because of an evolution of the adopted accessibility standards (e.g. WCAG 2.0) or because new devices are supported and new mobile OK tests are defined, the Assessment Tests, represented as a class in the figure above, are actually different

classes, one for each test. Integrating a new test means to add a new class, corresponding to the new available test, and register a new Guideline and its relevance condition. The figure 3.4 depicts an example of possible tests.

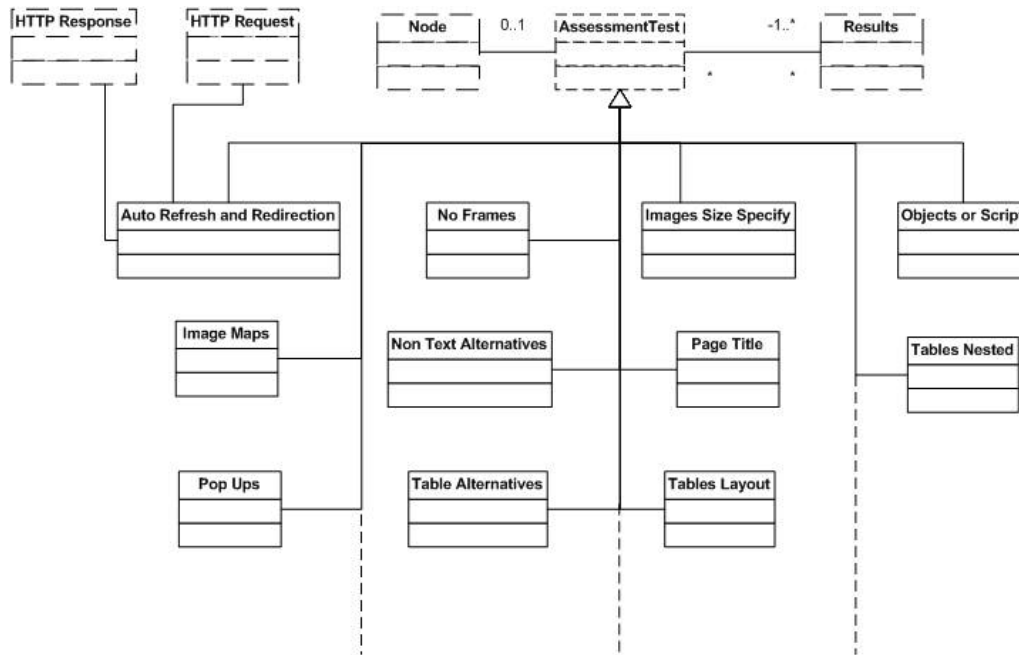


Figure 3.4: Tests Class Diagram

Even if the vast majority of tests fall upon HTML Nodes, the information about the HTTP request and corresponding HTTP response is also required by some mobile adequacy tests, and thus should also be available as obtained by the Resource Handler, through the MWAAAT class

The Results Handler manages the evaluation results and ultimately the results presentation. For each assessment request upon a chosen resource the handler collects a set of results. These are either results from an HTTP request/response assessment, a whole document verification or specific node checking.

3.4 Non-functional requirements

The MWAAT tool should take into consideration in its design and development a set of non-functional requirements derived from the Accessible project [21].

Performance requirements

- P-REQ1-1: Average system response time shall be in proportion with the complexity of the objects that are tested.

Operational requirements

- O-REQ1-1: Where appropriate system components shall be able to generate automated help wizards and error messages in case of system malfunctions and/or users mistakes.
- O-REQ1-2: Colours and contrasts shall be in accordance to usability and accessibility guidelines.
- O-REQ1-3: A help menu shall exist, that will facilitate the user understand the operation of the system, and guide him along the process.
- O-REQ1-4: All system components shall be easy to navigate to people with different knowledge and capabilities (developers, designers, testers, etc.)

Reliability requirements

- R-REQ1-1: Most of modules shall provide diagnostics messages in case of unsuccessful or uncertain operations.

Maintainability and Interoperability requirements

- M-REQ1-1: All software modules developed in the project will be released under an open source license.
- M-REQ1-2: All system components will be implemented in modular, open source system architecture
- M-REQ1-3: The framework shall be based on a layered solution with a high level of encapsulation of components to assure the maintainability of the infrastructure with future upgrades.

3.5 Summary

The chapter presented a further analysis of the application domain problem, considering the articulation of mobile adequacy and accessibility evaluation. Departing from the previous chapters' analysis of the state of the art, the relation between MWBP and WCAG guidelines was examined, not only in terms of overlapping conditions, already done by others, but also in terms of the complementary of their application. The disability type's influence on the selection of the MWBP guidelines was also scrutinized and the steps to find a resulting coherent guidelines subset was identified for each disability profile. Those include the clear identification of overlapping conditions on MWBP and WCAG guidelines; the consideration of accessibility independent MWBP guidelines and WCAG

mobile independent ones; and finally the exclusion of MWBP guidelines that may be irrelevant for specific disabilities.

The chapter follows through the specification of actors and use cases required for the creation of a mobile accessibility tool that handles disability profiles. Use cases are described providing main and alternative flows of events. Finally an overall class diagram is drawn and discussed and a set of non-functional requirements presented.

Chapter 4

Design

MWAAT software requirements have been analyzed and modeled in the previous chapter. We proceed in the current chapter with the software design, setting the stage for construction [27].

4.1 Architecture

Software architecture alludes to the overall structure of the software and the ways in which that structure provides conceptual integrity for a system. In its simplest form, MWAAT architecture defines the structure and organization of program components (modules) and the manner in which these components interact.

One of the main pillars of the design rationale, for MWAAT architecture and for the whole design process, was functional independence. This concept, an outgrowth of modularity and of the concepts of abstraction and information hiding, is fundamental in the MWAAT design process in order to ensure smooth future integration with the rest of ACCESSIBLE components. In fact, some of the modules are under development for different platform approaches (see section 2.6), and need to integrate different parts of MWAAT. As a result we conceived MWAAT modules minimizing interaction with each other.

MWAAT architecture comprehends five main modules: The MWAAT main module, the User Interface, the Selector, the Evaluator and the Results Handler (see Figure 4.1).

4.1.1 MWAAT

The MWAAT module is responsible for the initialization of all the other components, namely the GUI, the selector, the evaluator and the Results Handler. It does also provide the communication means between the User Interface and all the application domain modules (i.e., Selector Evaluator and Results Handler). As such, it ensures the primary level of functional independence that will allow the replacement of the GUI presented later in this document by another, such as the Accessible Portal (see section 2.6).

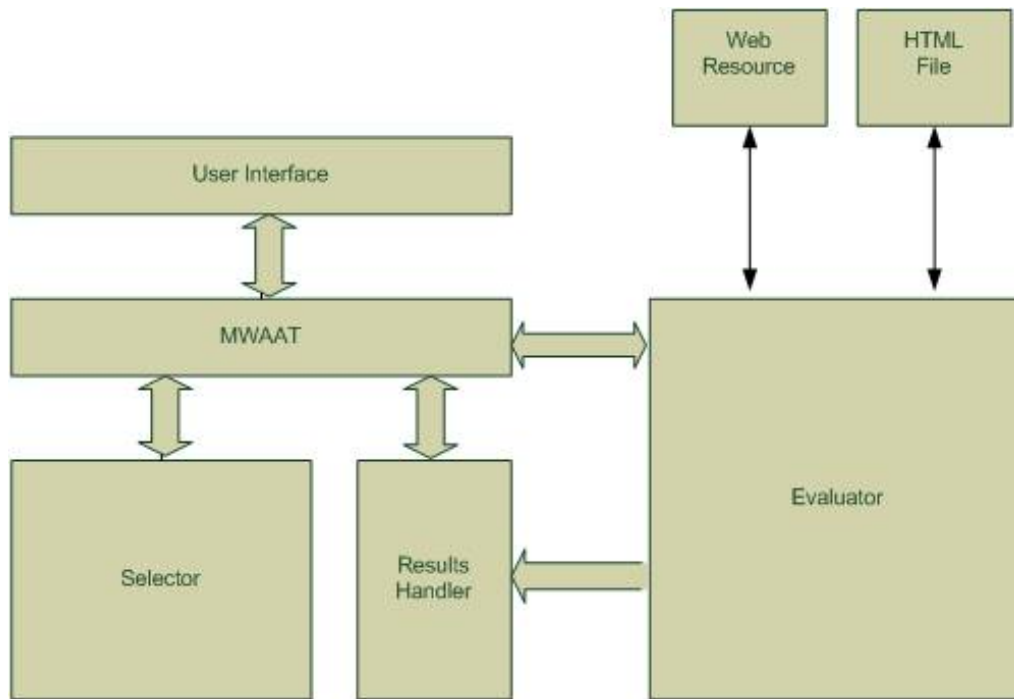


Figure 4.1: MWAAT Block Diagram

4.1.2 User Interface

The User Interface module may be a graphical user interface, a Web services interface or a command line interface. Its role is to allow: a) the input of the URL or the HTML file to be accessed; b) the setting of the Delivery Context, a mobile or a desktop access; c) the choice of the disability type; d) the choice of the device type and e) the set of generic parameters like proxy server and port and the presentation of results.

All user data and events gathered by UI will flow through the MWAAT module that will distribute them to the adequate domain module. Conversely, all data emerging from those modules that needs to be presented to the user, will reach the UI module through MWAAT. Thus, the UI constitutes an encapsulated thin layer, with minimal domain knowledge, that is easily replaceable.

4.1.3 Selector

The Selector module will have the knowledge about the disability types available, the device types offered and guidelines that are relevant to each usage scenario based on the user selected disability and device types. In general this module implements the proposed methodology.

As such, from the selected pair, device type and disability type, the Selector will send back to MWAAT the set of guidelines that apply to the selected usage scenario. Typically, as in the implemented solution, the UI will present this set to the user allowing

a comprehensive user feedback. Later, when the assessment request is issued by the user, the relevant guidelines' subset will be delivered to the Evaluator.

Again, with this arrangement, the Selector has no knowledge of the remainder modules. This fact, as it happens with the UI, permits an easy adaptation within two dimensions: the substitution of the Selector by a wider expert system, such as that developed in the project; the use of the Selector in other light versions of the tool, such as the command line based.

4.1.4 Evaluator

The Evaluator module will be responsible to access and evaluate the Web resources indicated by the user according to the set of determined relevant guidelines. Consequently, it has primarily to: a) disentangle the type of resource the user wants to assess, i.e., direct file access or HTTP request/response; b) set the current active list of tests, from the intersection of those available in the evaluator and those associated with the relevant guidelines set, previously determined by the Selector; c) parse the resource and associated ones (linked and embedded); d) apply the adequate tests to the document, the nodes and the associated resources; e) feed the evaluation results to the results handler.

Considering the data flow to and from the Evaluator, this module will essentially receive an evaluation request, including the resource to be evaluated and the set of relevant guidelines that will be used for that evaluation and provide the Results Handler with the evaluation results.

Here, again, the functional dependences are reduced to a minimum. It is worth noticing that the stronger connection to the Results Handler, and the design classes corresponding to the previously mention Resource Handler (see section 3.3) is a consequence of: a) the intrinsic relation between the assessment tests and the results they issue; b) the particular way some tests depend on how the Web resource was obtained (context delivery); c) the need to process, in some cases, the resources associated with the main Web target. As such, the design choice was to relax the functional independence requirements in order to gain in performance. Nevertheless, the adequate protocol between these classes waives the flexibility costs to be paid. In fact, for example, the introduction of a different evaluation results reporting format can be easily obtained either with a different Results Handler class or simply a change on the User Interface module.

4.1.5 Results Handler

Finally, the Results Handler module is responsible to gather the test results and provide the detailed evaluation results on demand. Its relevance on the architecture diagram results, again from the emphasis on module independence and on the ACCESSIBLE requirement of supporting different reporting styles (as just stated before).

4.2 From requirements to design

In this section we present the realization of the MWAAT main use cases using interaction diagrams. For use cases that directly derive from user actions, the related user interface that partially implements it is shown as the trigger of the corresponding interaction diagram. This approach will ease the connection to the implemented tool.

4.2.1 Use Cases Design

Departing from the use cases, the architecture and the adopted design decisions we identified four main areas for the presentation of the interaction diagrams: guidelines selection; resource access; results handling; and evaluation.

Selecting relevant guidelines

The figure 4.2 represents the sequence diagram that realizes the Set Relevant Guidelines use case as well as the Select Disability Type and the Select Device Type ones.

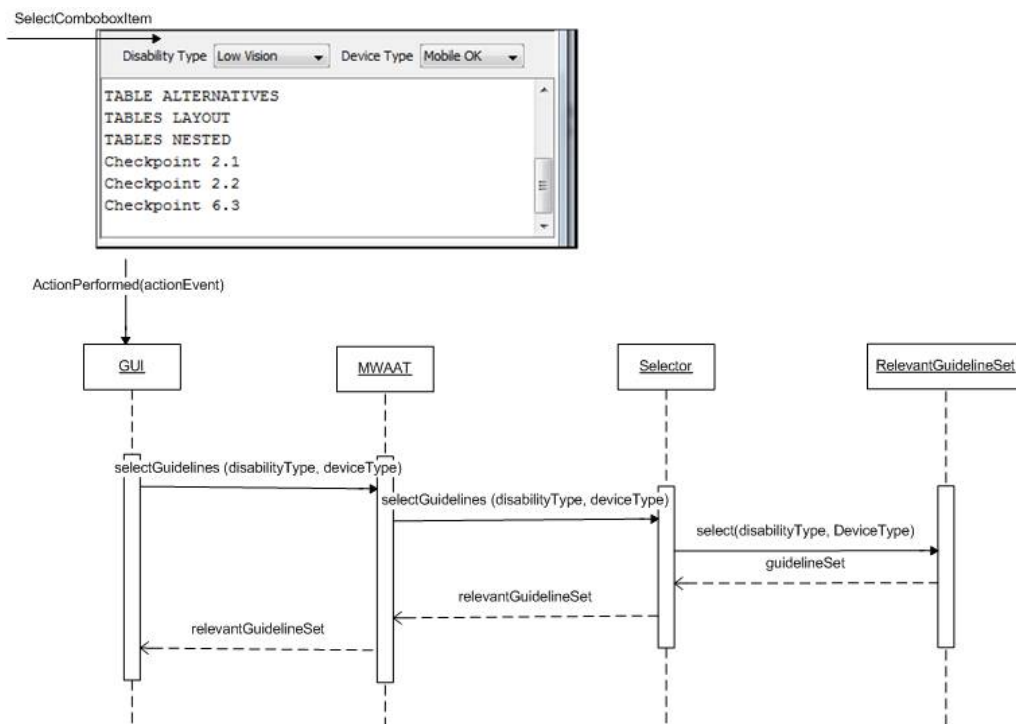


Figure 4.2: Select relevant guidelines

Whenever the user selects a value from either combo box, Disability Type or Device Type, the event triggered in the GUI requests a new set of relevant guidelines. The request follows with the selected disability type, device type to the Selector through MWAAT ensuring the aforementioned independence. The Selector then builds the relevant Guidelines Set according to the proposed methodology (see previous chapter) and returns it to the UI,

across the MWAAT instance. MWAAT provides the necessary transformations on the set to be compressible by the GUI or other UI that may substitute it. The final defined protocol establishes a set composed of a list of URI that refer to the specific guidelines, thus covering WCAG or MWBP.

The GUI is also responsible to show the relevant set to the user as denoted by the figure. Order, final aspect or even selection refinement operations are on the responsibility of the GUI, as well as keeping the adequate feedback to the user on which Disability and Device Types are selected.

Since the GUI must provide to the user the list of the available Disability and Device types, in order to maintain the coherence, that list is requested to the Selector that ultimately implements the guideline selection methodology. The figure 4.3 depicts those requests, again from the GUI to the Selector. These are typically issued as UI initializa-

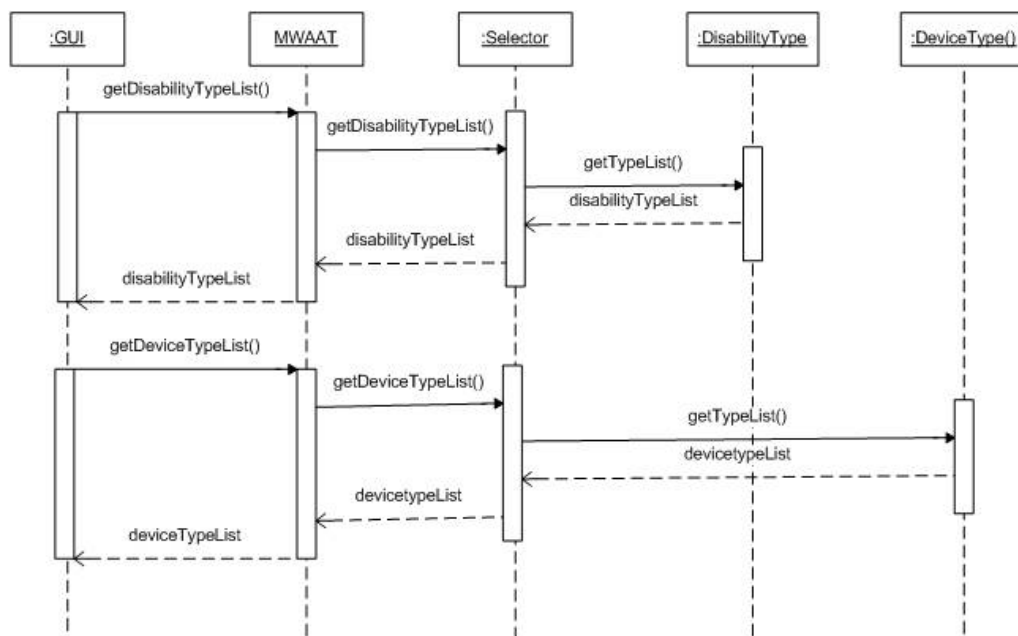


Figure 4.3: Disability type and device type initialization

tion occurs. It is thus important that the Selector is initialized before the UI. Interestingly, the Selector does not need to keep the state, whereas the UI keeps it intrinsically, in order to show it to the user. This will reinforce the independence of these two modules.

Accessing resources

The access to the Web resource that will be evaluated can be done through two main methods: direct file access and URL based. The former can be accomplished directly through the use of off-the-shelf components, like a file browser, and thus the corresponding use cases realizations are omitted here. The latter, implies further work and comprises the Set Proxy, Chose URL and Select Delivery Context use cases.

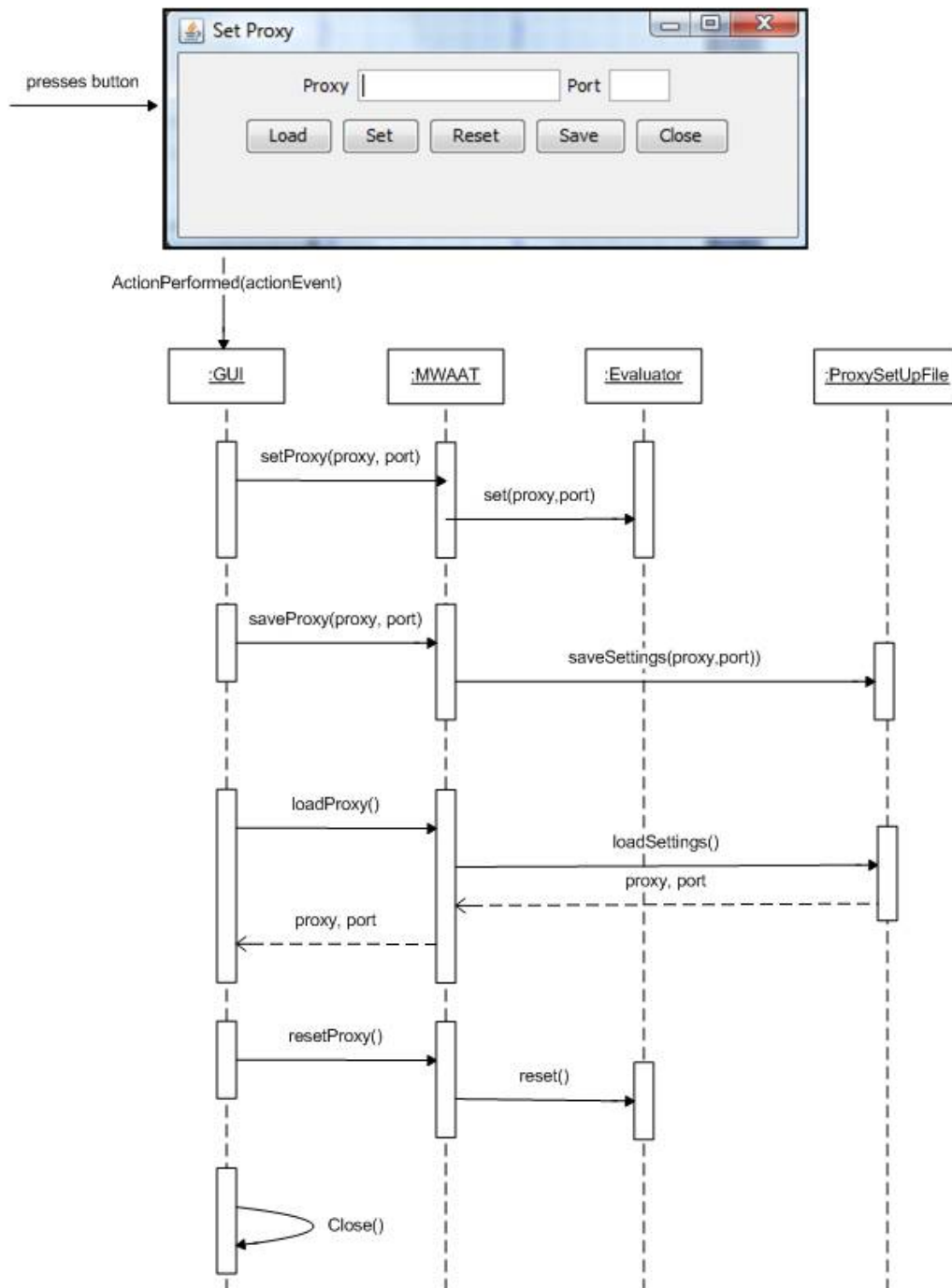


Figure 4.4: Set proxy

In figure 4.4 we present the Set Proxy sequence diagram and its associated graphical user interface design. This will insure that MWAAT works for networks where access to the Internet must be performed through a proxy server. The design is sufficiently flexible to support the usage of the tool in different network access settings.

Figure 4.5 illustrates the initialization of the delivery context list in the UI and its definition. The kept choice will enable the Evaluator to make the adequate request to the

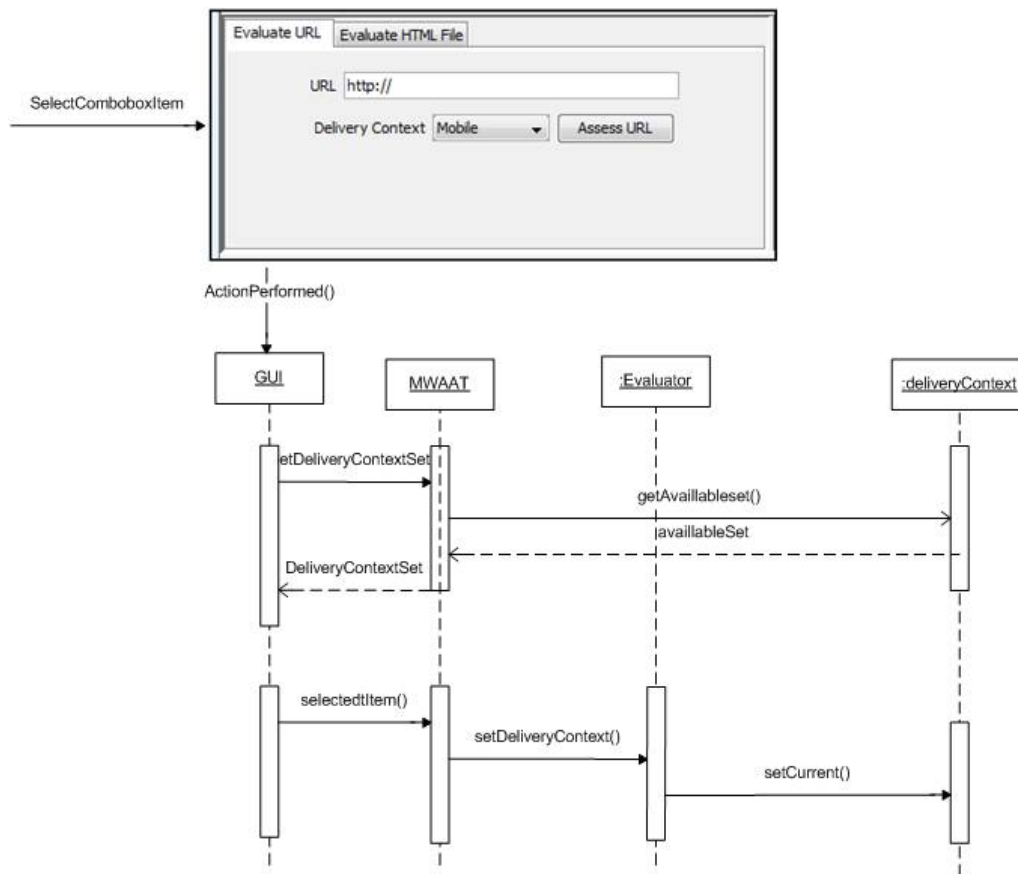


Figure 4.5: Select delivery context

Web server in order to obtain the adequate representation to be assessed. Moreover, it will allow the application of specific assessments tests that require the whole information.

In the figure it is also visible the realization of the Chose URL use case, which correspond simply to the URL specification, and in the hidden “tab” the access to direct file access one. The Assess URL button or other command issuing approach (e.g. the one on the hidden “tab”) will trigger the evaluation of the selected Web resource.

Handling Results

Handling results involves keeping them in order to respond to presenting results, presenting html code and to save results requests. Figure 4.6 shows the Handle Results use case sequence diagram. The requests reach the Results Handler in consequence of user actions, either explicit or implicit. Particularly, the presenting operation may be requested by the GUI, through MWAAT, by automatic UI refreshing needs, or because the user issued the command explicitly. The Node Printer will have the knowledge of handling the parser HTML tree of nodes and build a presentable HTML document.

Saving results can be issued through two different paths. Either the user defines a default saving results file, which will keep the results after each evaluation, or explicitly

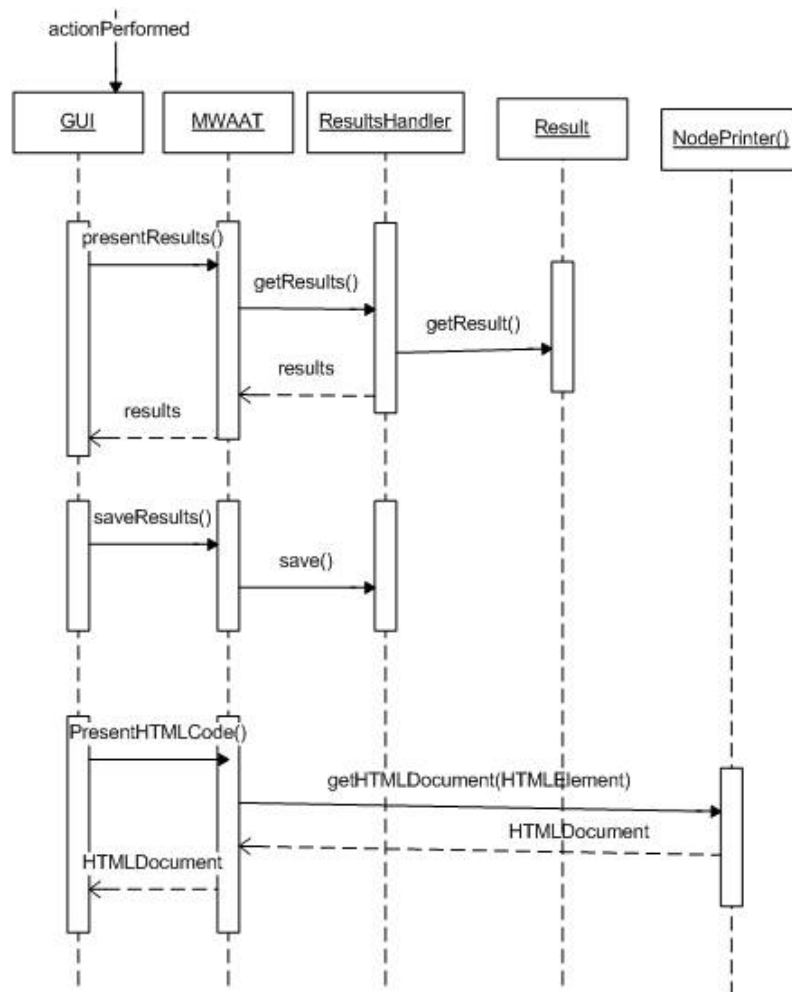


Figure 4.6: Handle results

saves it in a specified file. The later, again, corresponds to browsing and selecting a file and thus is a subset of the former. The configuration use case allows to set a default results filename as described in figure 4.7

The remainder cases of handling results roughly correspond to gathering the outcome of the several assessment tests and deliver them to the UI through MWAAT. Nevertheless, as previously stated, much of the interaction occurs within the evaluation process.

Evaluating resources

Evaluating resources comprehends two main use cases: Access Web Resources and Access Resource File. The figure 4.8 represents the interaction diagram for the first one.

As previously mentioned, the interaction is triggered by an access URL command, represented in the figure by the “Assess URL“ button. As usual, the GUI gathers the data required to execute the assessment and delivers it to MWAAT. MWAAT add whatever

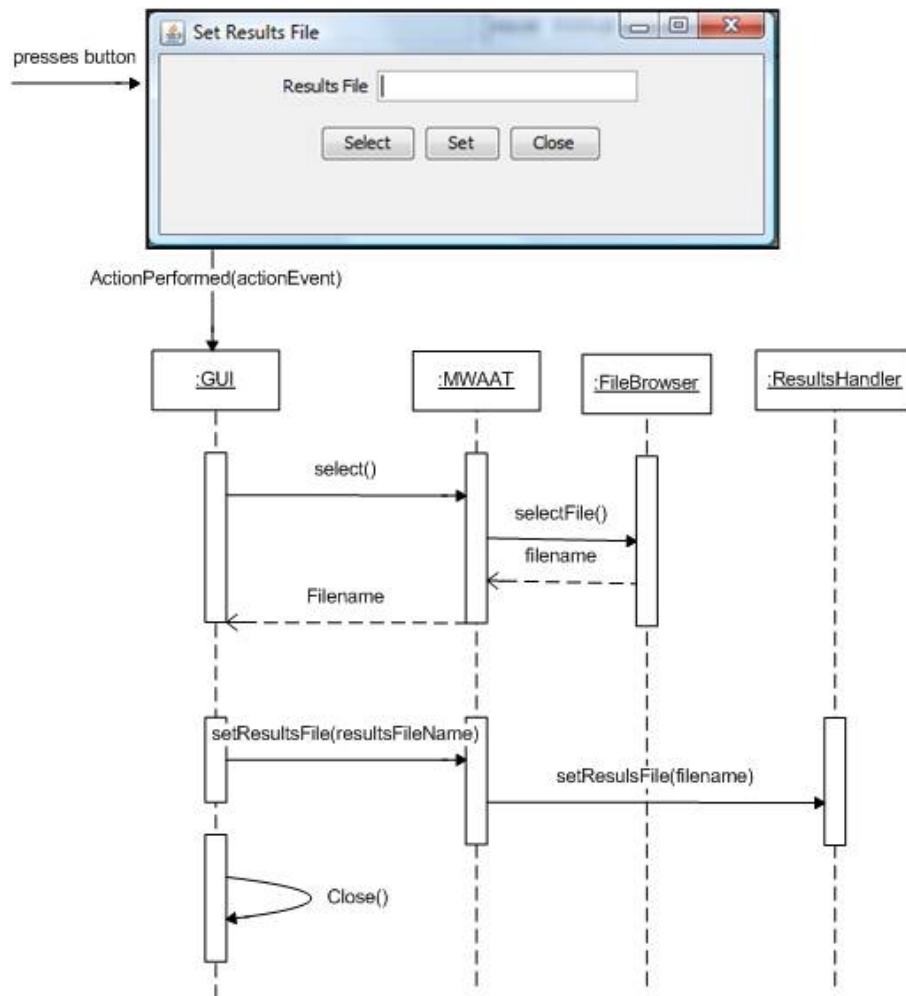


Figure 4.7: Results default filename setting

missing data is required, adjust it as needed and invokes the Evaluator. At this point the required data includes the resource's URL, and the set of relevant guidelines to be used in the assessment.

The Evaluator begins by obtaining the working objects from the received data. These include the assessment tests correspondent to the relevant guidelines (i.e., executable pieces of code instead of guidelines declarations) and the actual resource. The tests have a straightforward correspondence to the guidelines and are obtained by a simple fetch operation. Acquiring the actual resource to be evaluated requires the use of the previously set delivery context and the current proxy setting. With these two, plus the receive URL, the Selector performs an HTTP request to the Web Server, finally obtaining the resource on a successful HTTP response.

Afterwards the Evaluator will ask the parser to process the HTML document, obtained within the HTTP response content, receiving the HTML parsed tree of document nodes. It will then be able to apply the current relevant tests to the HTTP response contents and to each of the document nodes as illustrated in figure 4.9.

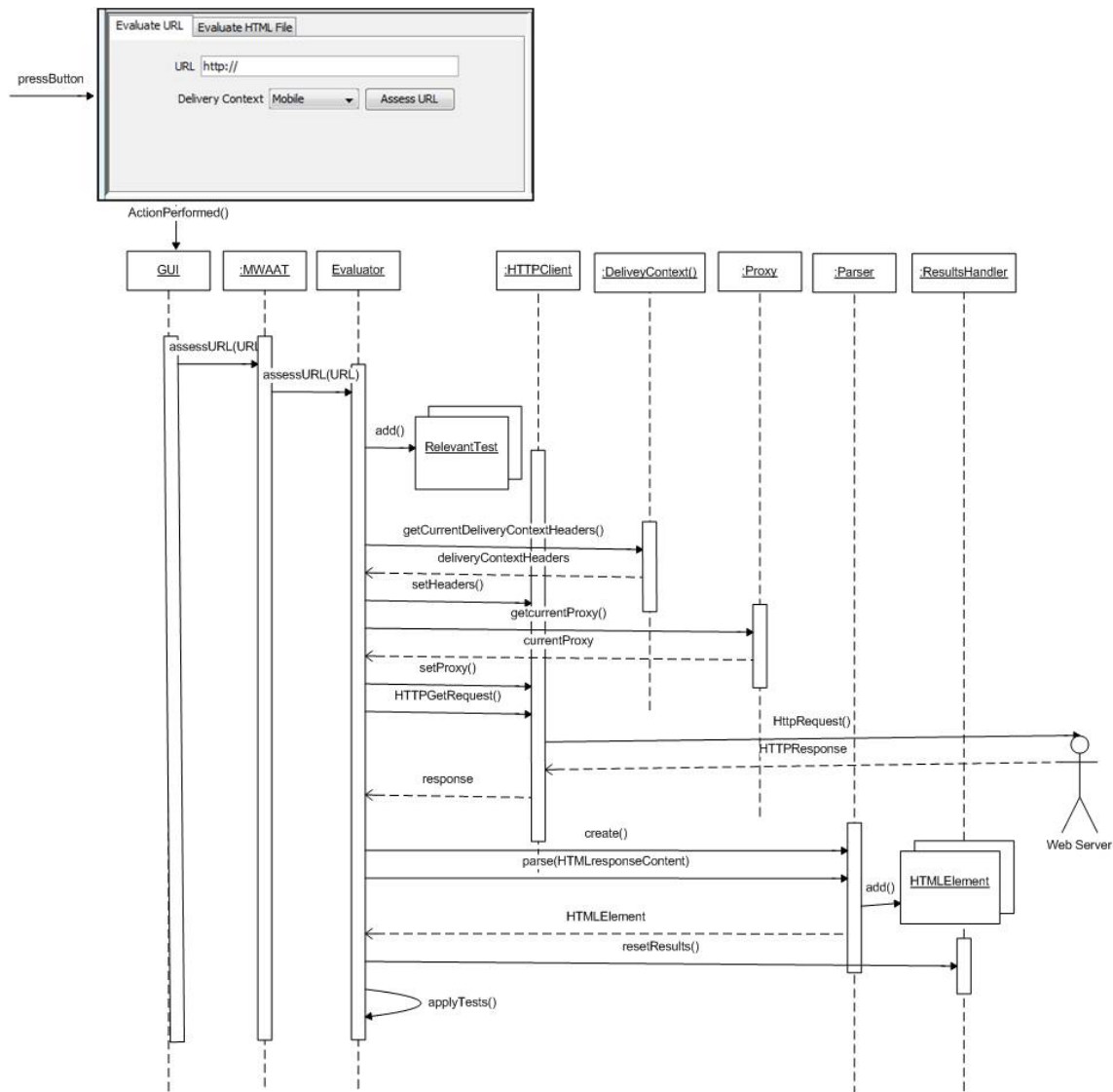


Figure 4.8: Assess URL - resource handling and parsing

The relevant document tests that apply to the general HTTP response contents and do not depend on the particular node contents are first applied (see figure 4.9). An example is the content format support test that assesses whether the document's Internet media type, as specified in the HTTP response Content-Type header, is "application/xhtml+xml", "application/vnd.wap.xhtml+xml", or "text/html".

Afterwards each HTML document node is assessed on a depth first algorithm order, by each and every one of the relevant node tests (see figure 4.9). Each test will assess the compliance of the node contents with the conditions it should comply with, and has inside the knowledge of the preprocessing information he needs to assess.

Some node tests only evaluate HTML related information, such as the table alternatives test that looks for the existence of table node names. Other tests evaluate not only HTML related information but also HTTP request and HTTP response related informa-

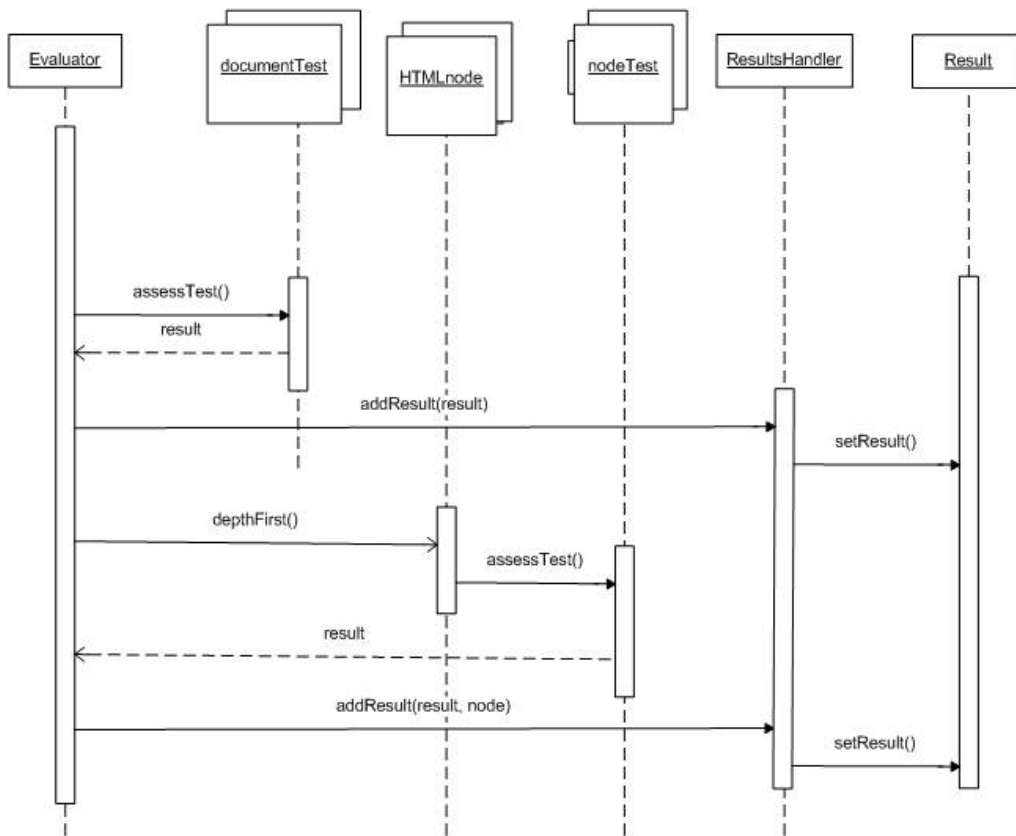


Figure 4.9: Assess URL - test evaluation

tion. The auto-refresh and redirection test for example, looks for <meta> HTML nodes with “HTTP-equiv” attributes content of “refresh”, and also looks for HTTP response contents with a different URL than the one sent on the HTTP get request. Each node test will receive a node to evaluate and will be responsible to evaluate its contents against the needed information such as HTTP request and/or the HTTP response or the CSS style information for example.

The use case for Access Resource File is a simplified version of the above. It maintains the steps for a) getting the tests; b) parse the HTML tree; and c) apply the adequate tests. However, the acquisition of the resource is based on a file browser. The file is opened and parsed before the tests are applied.

It is the Evaluators responsibility to know that the assessment under action is an URL or a file evaluation. Each tests knows what evaluating conditions are applicable or not in each assessment scenario. When accessing a resource file only the HTML related information is evaluated, since there is no HTTP request and response contents to be evaluated.

4.2.2 Design Class Diagrams

In the previous chapter in the analysis domain class models we defined a complete set of analysis classes. Each of these classes describes some element of the problem domain, focusing on aspects that are user visible, at a high level of abstraction.

At the design level, we define a set of design classes that refine the analysis ones by providing design detail that enable the classes to be implemented. Moreover, we create where needed a new set of design classes that implement a software infrastructure to support the solution.

In this design class diagram (figure 4.10) we present the main methods derived from the interaction diagrams previously described.

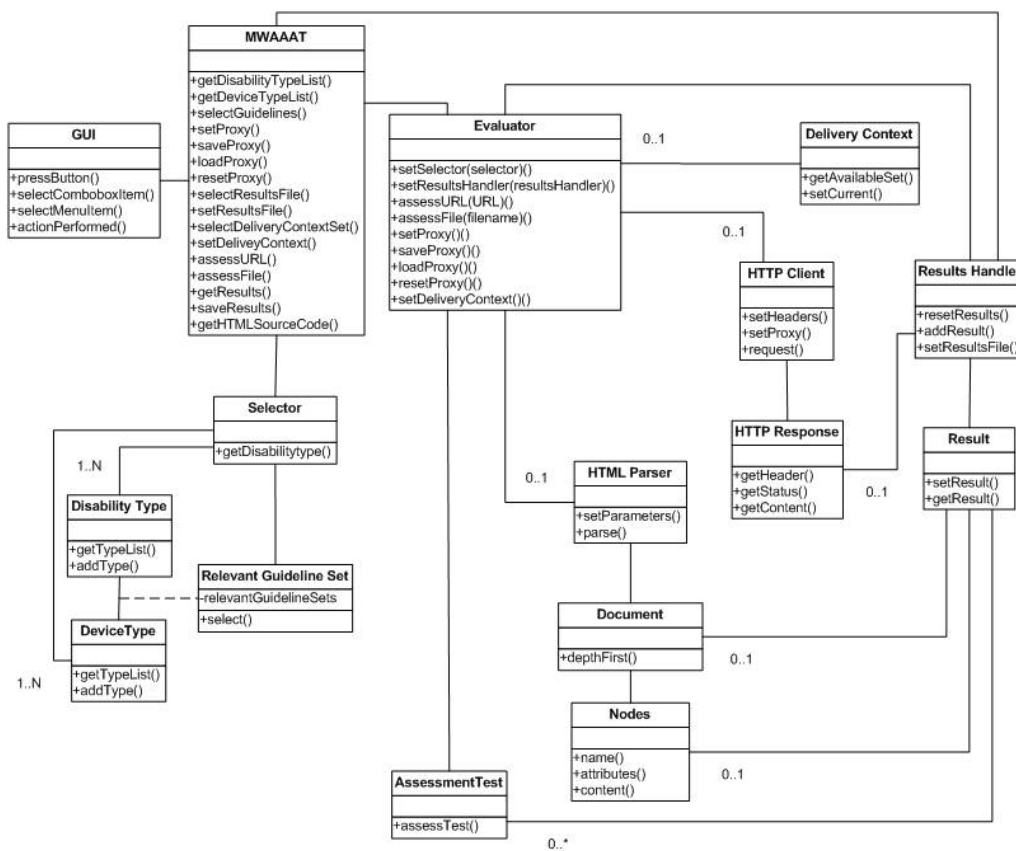


Figure 4.10: Design Class Diagram

Particularly relevant are the classes for the assessment tests. The diagram in figure 4.11 attests some of these classes, in fact, the subset of all the possible WCAG and MWBP guideline that were selected for implementation. The choice was based on criteria that enabled us to show the added value of the tool when evaluating different usage contexts.

As referred before, each new test is represented by a new class. Adding a new test is attained by a new subclass that refines method `assess()`, as seen in the figure. Each of

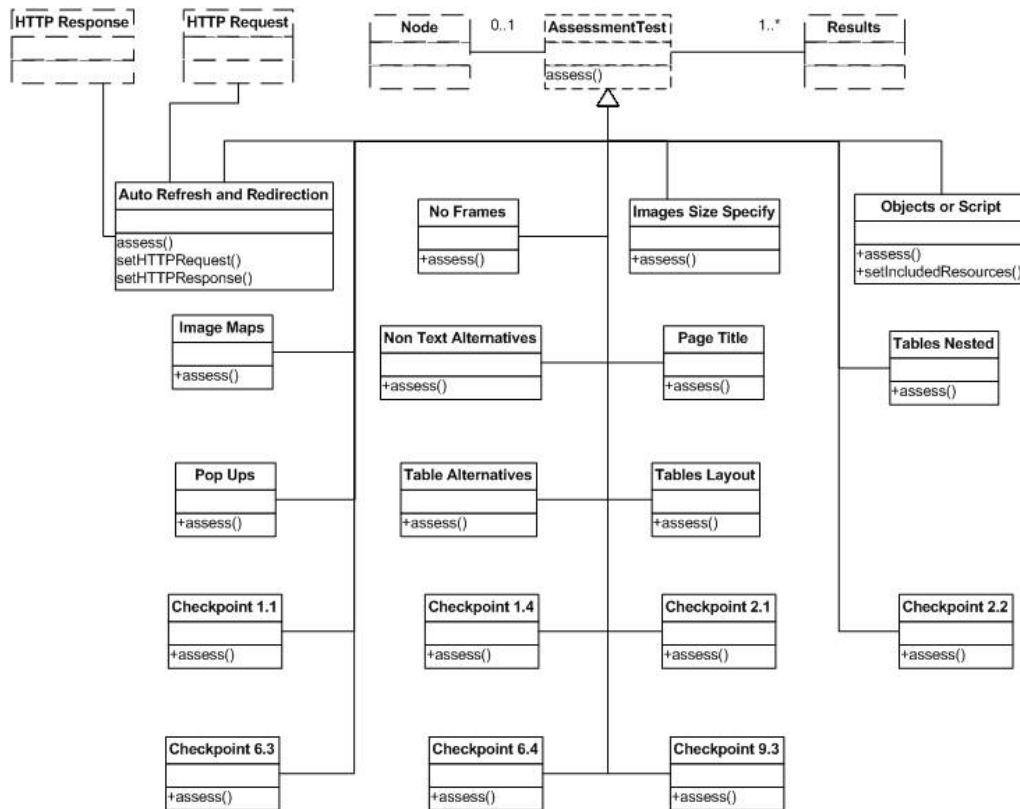


Figure 4.11: Mobile adequacy tests class diagram

the tests will have access from the results handler and the working Node inherited from the AssessmentTest class it extends. Fetching further information is the responsibility of each test implementation. In fact, AutoRefresh and Redirection is an example of that (in the figure), needing to access HTTP request and HTTP response information in order to execute its evaluation conditions.

4.3 Summary

This section presented the main design considerations of MWAAT. The architecture was described and interaction diagrams were presented and discussed for the most relevant use cases defined in the previous chapter. The chapter end revising the main design classes.

Chapter 5

Implementation

In this section MWAAT's interface and the major implementation options, are described. After a brief description of the development environment each MWAAT architecture component is presented, emphasizing the most relevant details of the implementation.

5.1 Environment

MWAAT was developed using Groovy, an agile and dynamic language for the Java Virtual Machine. Groovy provides some advantages, by making Java code simpler to write. It automates recurring tasks and supports ad-hoc scripting. Groovy also offers language features such as closures or dynamic typing. Overall, code based in this language can be made simpler to read [37].

Considering the requirements of a tool such as MWAAT, involving Web based documents' manipulation, strong flexibility and easy adaptation to other developed modules, and of course the prerequisite to become open-source code, high level Web constructs, readability and ease of maintenance are important features. Groovy provides this support.

5.2 Graphical User Interface

In figure 5.1 MWAAT graphical user interface is shown. As expected the access to “save results” is available through the file menu. The **View** menu controls what is shown in the lower pane, permitting to alternate between results and source code. The panes above the lower one are responsible for the selection of the resource to be evaluated (left) and the choice and visualization of the set of guidelines that will be applied (right). Finally, the **Tools** menu offers mechanism to configure the tools, such as the **Set Proxy**, for which the UI can be seen in the previous chapter.

In general, the created user interface follows the current practices for menu organization. Regarding panes, a left-right, top-down approach was adopted. Usually, the user first selects the resource to be evaluated (left pane). Then, for different combinations of guide-

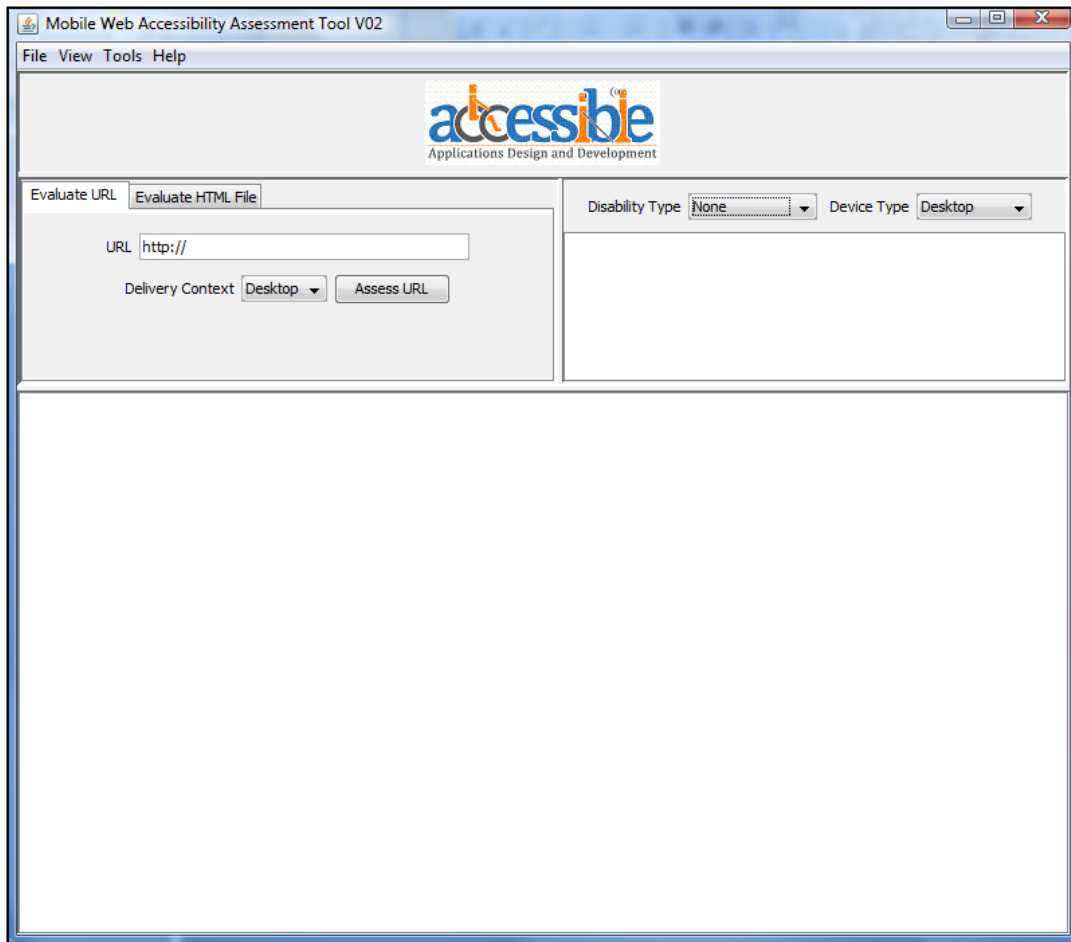


Figure 5.1: MWAAT graphical user interface

lines (right pane), he/she inspects the evaluation results (lower pane). The ACCESSIBLE logo follows from project's policy.

Figure 5.2 shows the use of a file browser here as a consequence of a save results issued command. The same File Browser is also used whenever a file or directory needs to be referred, for example, when the user opts to evaluate a Web resource file, not yet deployed on a Web server.

Figure 5.3 depicts MWAAT left pane, showing the “tab” that allows the user to make that selection (the **Select File** button). It can also be seen that the pane includes two “tabs”, **Evaluate HTML File** presented in the figure, and the **Evaluate URL** shown above on figure 5.1. Both tabs provide an **Assess** button that triggers the evaluation process. The **Evaluate URL** tab allows the choice of the **Delivery Context** through a corresponding combo box.

Two other combo boxes, available on the right pane, are used: **Disability Type** and **Device Type**. The choice of this user interface object allows the user to have the notion of the parameters applied in the evaluation and still relinquish space for the results pane. Moreover, since each one is dynamically initialized it allows an easy extension of the

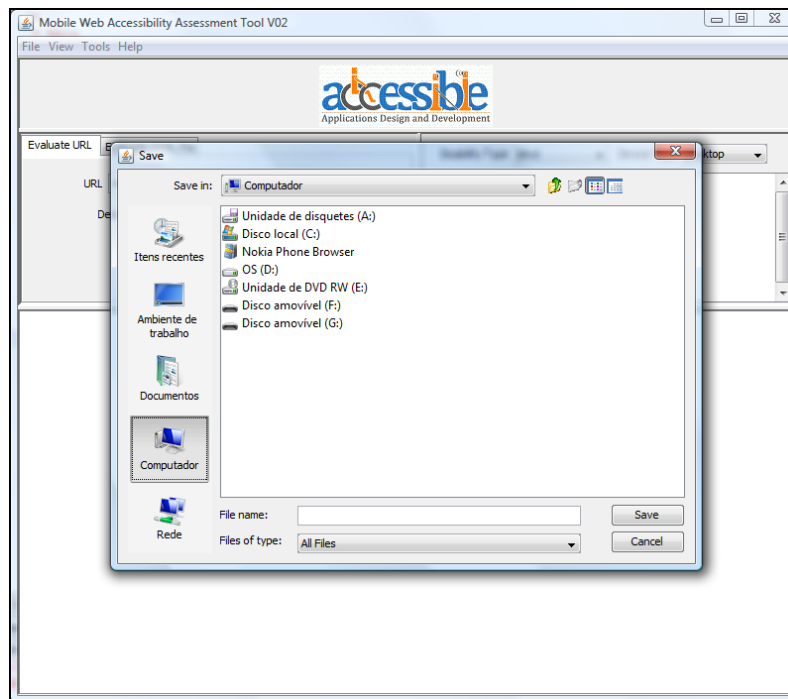


Figure 5.2: Save Results user interface

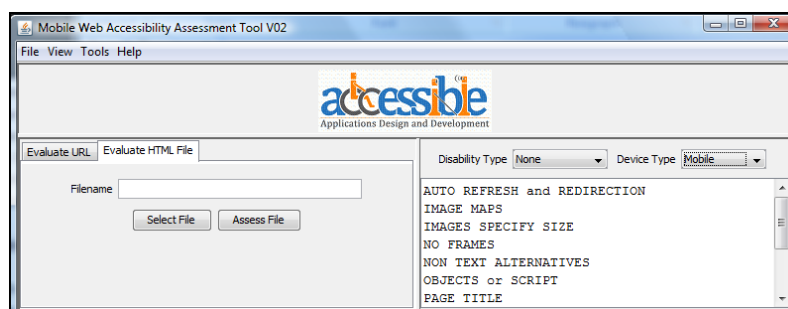


Figure 5.3: Evaluate HTML GUI

tools' capabilities in terms of new delivery contexts, possible refinement of disability types and other more specialized device types. Of course, as a list it allows the user to select only the supported variants.

The GUI implementation code uses javax swing and groovy swing builder. Swing builder allows creating full-fledged Swing GUI's in a declarative and concise fashion. It accomplishes this by employing a common idiom in Groovy, builders. Builders handle the busywork of creating complex objects, such as instantiating children, calling Swing methods, and attaching these children to their parents. As a consequence, the obtained code is readable and maintainable, while still allowing the access to the full range of Swing components.

5.3 MWAAT

The architectural component MWAAT comprehends the MWAAT system main class. It is responsible to create the other MWAAT main architectural objects such as the selector, the evaluator, the results handler.

MWAAT main class also acts as the system event handle component receiving all the events passed from and to the graphical user interface.

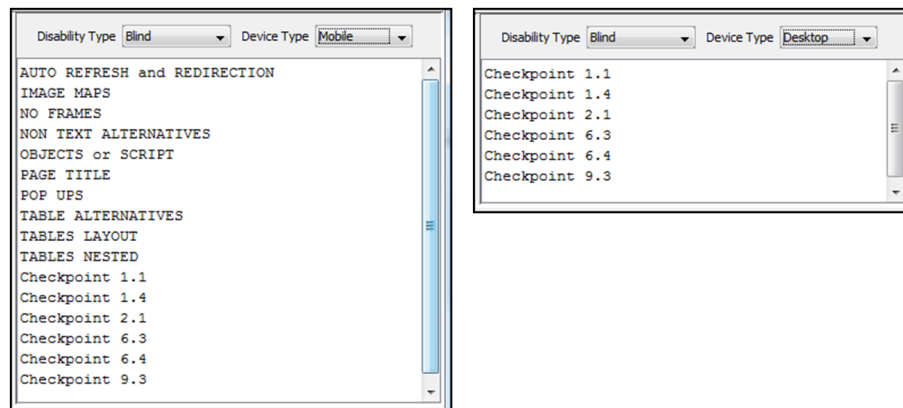


Figure 5.4: Blind disability associated guidelines

5.4 Selector

Figure 5.4 shows the listing of the supported guidelines relevant to the blind disability type, for a mobile accessibility evaluation (on the left) and for a desktop accessibility evaluation (on the right).

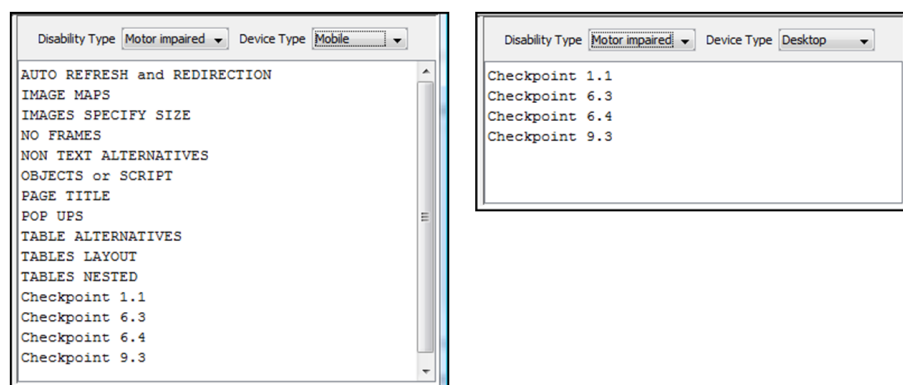


Figure 5.5: Motor impaired associated guidelines

Figure 5.5 shows the listing corresponding to a motor impaired disability for the same device settings

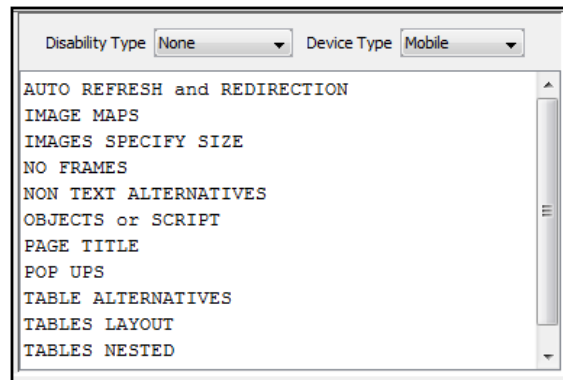


Figure 5.6: Mobile content guidelines

Finally, figure 5.6 depicts the relevant guidelines set when only a mobile adequacy evaluation is pretended. In this scenario no specific disability type requirements and associated relevant guidelines are used. The guidelines all respect to the mobile OK basic tests derived from the mobile Web best practices.

In each case, the sets of presented guidelines reflect the ones delivered by the Selector in consequence of the chosen parameters. The selector module, which can be easily replaced by the Accessible inference engine in other planned settings, is currently implemented as a Groovy class. The sets of relevant guidelines are supported as Groovy map collections.

5.5 Results handler

The results handler module implements two main classes for handling results and the evaluated intermediate HTML source code. This groovy class supports the needed methods to add the results obtained from each of the tests and the methods needed to return the results lists whenever needed.

They are implemented as Groovy lists of lists retrieving each of the HTML source code nodes that have associated fail or warning test results and each of the test fail or warning results.

The evaluated source code can be recovered and listed whenever the user needs it. Groovy's XML streaming markup builder and XML node print utilities in order to parse, build and print the source code listing.

5.6 Evaluator

The Evaluator module performs the following main tasks that should be emphasized here: HTTP request/response, HTML parsing, and assessment test execution.

5.6.1 Negotiating resource representations

Currently, MWAAT supports two delivery contexts namely the default representation access from a desktop and the mobile OK delivery context access. Extensions to other should be straightforward, since the Evaluator already supports several features of the HTTP content negotiation.

Supporting content negotiation as described in the HTTP/1.1 specification enables user-agents to acquire from the HTTP Server, the representation of a resource that best fits the browser-supplied preferences for media type, languages, character set and encoding. For example, for a given URL, a browser could indicate that it would like to see information in French, if possible, else English will do. This information is indicated in the headers of the HTTP request. The server will respond with an actual representation, that either corresponds to request's preferences (in French, for the example), or to the most close default (the English version). Web servers fully support, for example, the Accept, Accept-Language, Accept-Charset and Accept-Encoding request headers.

If multiple representations are available, the resource is referred to as negotiable and each of its representations is termed a variant. The ways in which the variants for a negotiable resource vary are called the dimensions of negotiation.

On the Evaluate URL tab (see figure 5.1), the user might choose the mobile delivery context or the desktop delivery context. This option sets the headers that will be used on the HTTP request.

Although the `java.net` package provides basic functionality for accessing resources via HTTP, it doesn't provide the full flexibility or functionality needed by many applications. The `HttpClient` class fills this void by providing an efficient, up-to-date, and feature-rich package implementing the client side of the most recent HTTP standards and recommendations. It is commonly used to build HTTP-aware client applications such as Web browsers, Web service clients, or systems that leverage or extend the HTTP protocol for distributed communication. It was therefore an obvious choice for MWAAT HTTP handling operations.

Complementing the above class, `HTTPBuilder` provides a convenient builder API for complex HTTP requests. It is built on top of Apache `HttpClient`, with Groovy syntactic on top. It also builds and parses common content-types, handling common content-encodings, and built-in support for common authentication mechanisms. `AsyncHTTPBuilder`, a subclass of the base `HTTPBuilder` which transparently delegates all requests to a thread pool for execution, is used in HTTP requests' handling.

5.6.2 Parsing HTML

The HTTP response contents or the HTML file source code HTML document is parsed in MWAAT using the `NekoHTML` parser.

NekoHTML is a simple HTML scanner and tag balancer that enables to parse HTML documents and access the information using standard XML interfaces.

NekoHTML enables to parse HTML documents and access the information using standard XML interfaces. Besides it can scan HTML files and "fix" common mistakes made while writing HTML documents: e.g., adds missing parent elements; automatically closes elements with optional end tags; and can handle mismatched in line element tags.

On the application we can set a variety of NekoHTML settings to more precisely control the behavior of the parser.

These settings can be set directly on the supplied parser classes by calling the `setFeature` and `setProperty` methods. In MWAAT we use the parser `setProperty("http://cyberneko.org/html/properties/names/elems", "lower")` in order to set all HTML elements to lower case. This property specifies how the NekoHTML components should modify recognized element names. Names can be converted to upper-case, converted to lower-case, or left as-is. We also use parser property's value of "match" to specify that element names are to be left as-is but the end tag name will be modified to match the start tag name. This is required to ensure that the parser generates a well-formed XML document.

5.6.3 Executing the assessment tests

The parsed MWAAT HTML document is evaluated against all the current relevant test that apply to the settled usage scenario using a depth first scan of the entire document and applying each of the tests to each of the nodes.

The relevant document tests that apply to the general HTTP response contents and do not depend on the particular node contents are first applied. An example is the previously referred content format support test that assesses whether the document's Internet media type, as specified in the HTTP response Content-Type header, is "application/xhtml+xml", "application/vnd.wap.xhtml+xml", or "text/html".

Then each parsed document node is evaluated on a depth first algorithm order, by each and every one of the relevant node tests.

Each test will assess the compliance of the node contents with the conditions it should comply with, and has inside the knowledge of the preprocessing information he needs to assess.

Some node tests only evaluate HTML related information, searching for the existence of certain node names and or node specific attributes or for specific occurrence of certain node attribute contents.

Other tests evaluate not only HTML related information but also HTTP request and HTTP response related information or CSS related information. Each node test receives the node under evaluation and is responsible for evaluating its contents against the needed information such as HTTP request and/or the HTTP response or the CSS style information for example.

Each mobile accessibility test implementation uses GPath to assess the node content and produce the evaluation results (Gpath is a means to select parts of the whole document).

	Gpath Query
node name	<code>if (node.name() == "meta")</code>
node attributes existence	<code>if (node.attributes(). "http-equiv" != null</code>
node attributes contents	<code>if (node.@ "http-equiv" == "refresh")</code>

Table 5.1: HTML node content query examples

On table 5.1 some examples of GPath queries are presented. On the first the occurrence of a `< meta >` HTML tag is evaluated. On the second example an “http-equiv” attribute existence is looked for and on the third example with the “@” symbol the condition tests the content of the “http-equiv” attribute.

HTTP response contents evaluation
<code>if(resp.getFirstHeader('Refresh')! = null)</code>
<code>if(resp.getFirstHeader('Refresh').contains("url = "))</code>
<code>def textList = resp.getFirstHeader('Refresh').tokenize(";")</code>
<code>if(textList[1] - "url =" != request.getURL())</code>
<code>nodeResult = "FAIL"</code>
<code>testMessage = "FAILon3.1AUTOREFRESHandREDIRECTION"</code> <code>-RefreshHTTPheaderisresentwithadifferentURI"</code> <code>"currentURLis\$request.getURL()</code> <code>redirectiontoURL = \$textList[1] - "url ="</code>

Table 5.2: An evaluation example

On table 5.2, an example shows HTTP response headers evaluation, and the comparison of the URL values sent on the HTTP request versus the one received on the HTTP response. The HTTP response is scanned searching for the existence of “Refresh” headers and if it is found, its content is evaluated in order to see if it contains an URL value. When this is true, the HTTP response URL value is compared with the HTTP request URL that was sent, in order to detect that the refresh corresponds to a redirection.

Each of the positive tests’ warning and failure results are added to the results handler. The inherent test result such as Warning or Fail are established and the messages corresponding to the applicable test conditions that were found are send to the results handler

to be stored attached to the HTML document node that is being evaluated. In the current version of MWAAT results are being stored as list of list using Groovy collections.

5.7 Summary

This chapter presented some of the most relevant details of the implementation of MWAAT. First the development environment was briefly presented followed by a description of the most relevant aspects of each of the MWAAT' modules.

Chapter 6

Results

In this chapter we present an example of a Web content development scenario and three evaluation case studies that illustrate MWAAT use and results. In the first example, MWAAT is used to help the developer reach a mobile accessible content, for an HTML file not yet deployed in the Web Server. This example serves mainly to show the use of the tool.

On other three case studies we access different Web resources, simulating the default desktop and the mobile delivery contexts and we evaluate the received Web content representations on the following Web resource evaluation scenarios:

- Default representation
 - accessibility evaluation specifying the disability type namely:
 - * all disability types
 - * blind disability type
 - * deaf disability type
 - * color blind disability type
 - * motor impaired disability type
- Mobile representation
 - mobile adequacy evaluation (no disabilities)
 - mobile accessibility evaluation specifying the disability type, namely:
 - * all disability types
 - * blind disability type
 - * deaf disability type
 - * color blind disability type
 - * motor impaired disability type
- Default representation

- mobile adequacy evaluation (no disabilities)
- mobile accessibility evaluation specifying the disability type, namely:
 - * all disability types
 - * blind disability type
 - * deaf disability type
 - * color blind disability type
 - * motor impaired disability type

6.1 Web content development example

In a developing environment situation, the developer's HTML source code is available on a file before deploying it on a Web server. Figure 6.1 presents a HTML source rendered in a desktop browser.

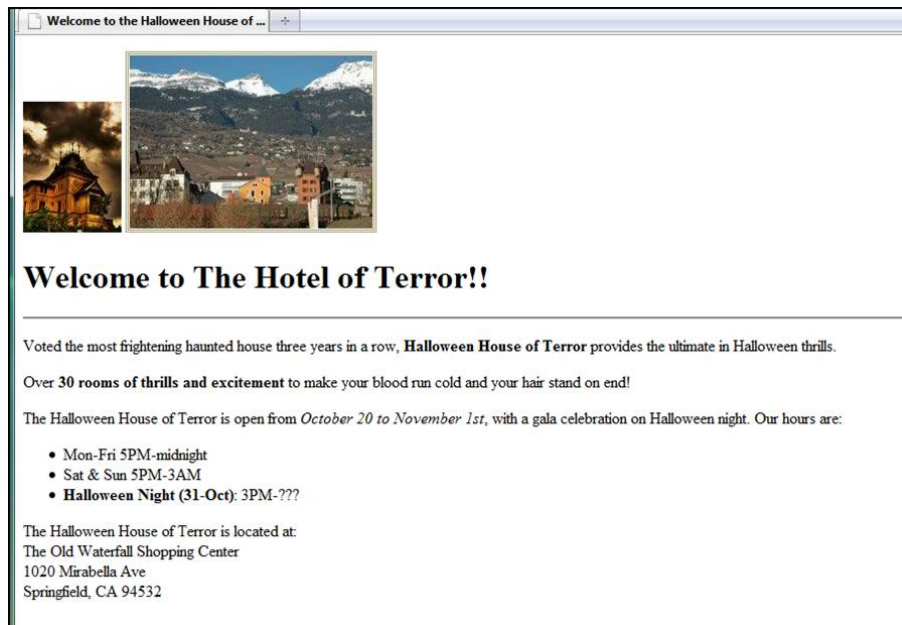


Figure 6.1: Web Content Example

With MWAAT the developer can evaluate if this Web content is adequate to be accessed from mobile devices or if there are issues to be solved. After specifying the filename on the adequate box and setting the device type to MobileOK, the developer will hit the AssessFile button. Figure 6.2 shows the assessment results regarding this Web content adequacy for mobile devices access. It is worth noting that the Web content source code presents a couple of issues regarding its mobile adequacy:

- One `` tag not only does not specify its size but also does not have an alt text attribute.

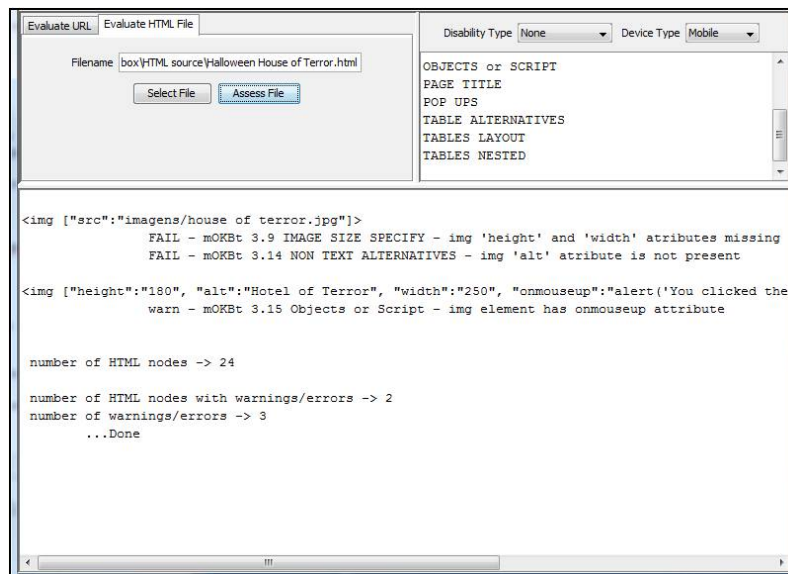


Figure 6.2: Mobile devices content adequacy

- Another `` tag has an `onmouseup` attribute which is not supported by many mobile devices.

Now consider that the developer wants to assess the content bearing in mind the accessibility regarding a motor impaired user, on top of a mobile adequacy scenario. He/she will change the settings on the Disability Type drop box accordingly. Figure 6.3 shows the corresponding results. Naturally, the same issues reported on figure 6.2 are also reported here. However, further issues are listed: those referring to the adequate WCAG guidelines.

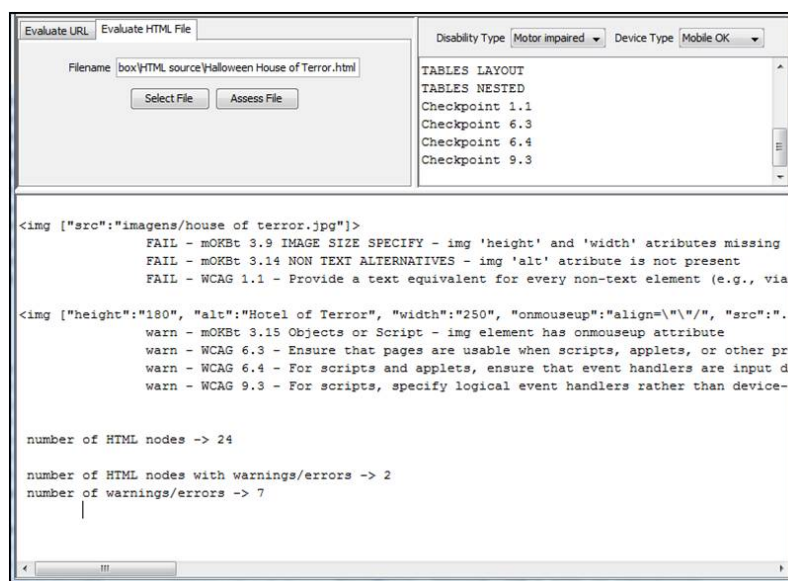


Figure 6.3: Mobile devices accessibility for motor impaired users

Changing the accessibility type to blind users is straightforward. We can see from

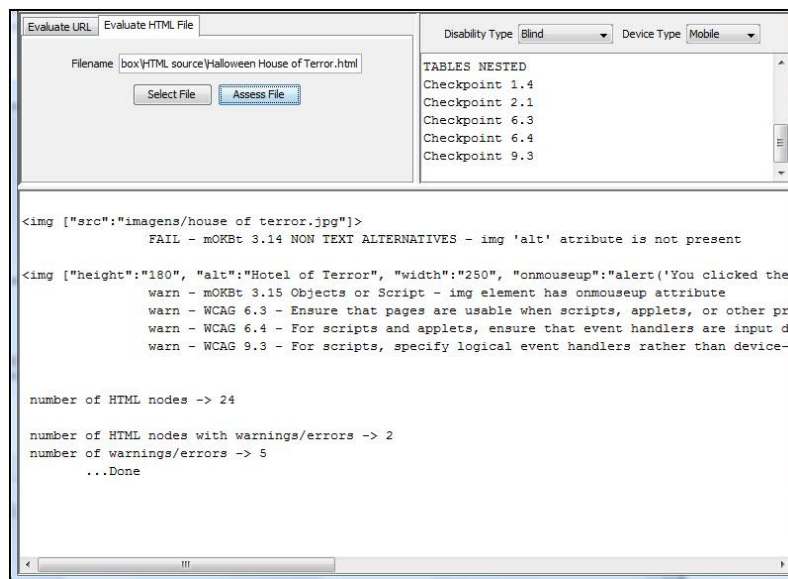


Figure 6.4: Mobile devices accessibility for blind users

figure 6.4 that although the HTML elements are the same, the impact is different since the images size specification is not relevant for blind users access according to HAM.

Once the issues are identified, the developer can solve them. The figure 6.5 shows MWAAT aspect when the user selects View Source Code option. Note that diagnostic

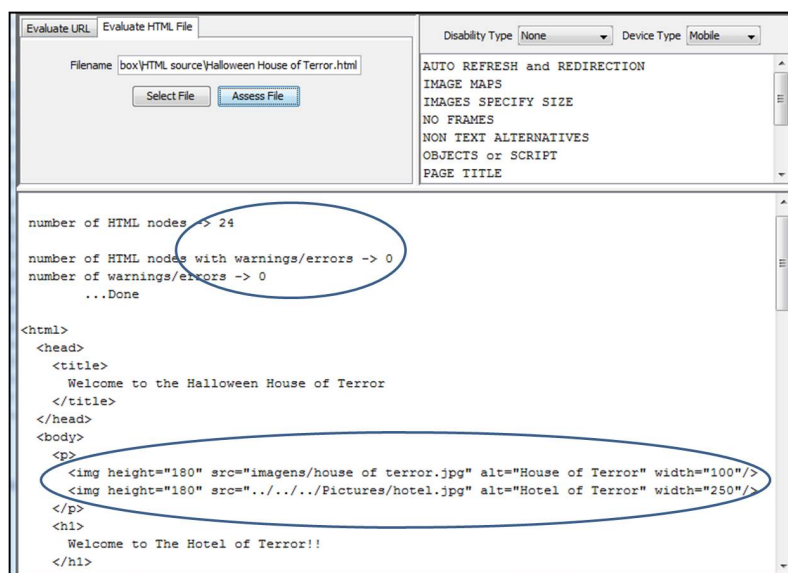


Figure 6.5: Mobile devices content adequacy and listing source code

presented in this figure lists the same number of nodes with no warnings/errors and that the HTML nodes that were reported in the results' panel of the previous figures are corrected.

6.2 Case studies

Three case studies are presented. All of them offer mobile representation alternatives and are sites with high rates of access. The first case study is used to illustrate more deeply the usage of MWAAT and the type of detailed information the developer will get from the assessment operations. Results for all cases are aggregated and discussed briefly in the following section.

6.2.1 Web Portal

The first case study is the access and evaluation of the Portuguese portal from the incumbent telecommunication company, Portugal Telecom, the SAPO portal (at <http://www.sapo.pt/>), regarding the information needed by developers. Offering services to mobile usage it is expected that it provides quality Web access at the mobile level.



Figure 6.6: Sapo Home Page

We started to access the SAPO portal URL “www.sapo.pt”, without specifying any specific content negotiation headers. Its home page snapshot is presented on figure 6.6. In figure 6.7 we can see the results from the evaluation of the URL default representation for an unspecified disability type. Most of the existing Accessibility tools perform this type of evaluation for WCAG rules conformance. The URL content representation received in figure 6.7, consists of an HTML document structure with 1453 HTML nodes, which would be not at all appropriate to receive and render on a mobile device and will easily exceed the page size limit guideline recommendation. Regarding accessibility, even considering the subset of implemented WCAG guidelines, MWAAT finds 71 document nodes with warning or error situations that should be analyzed and fixed.

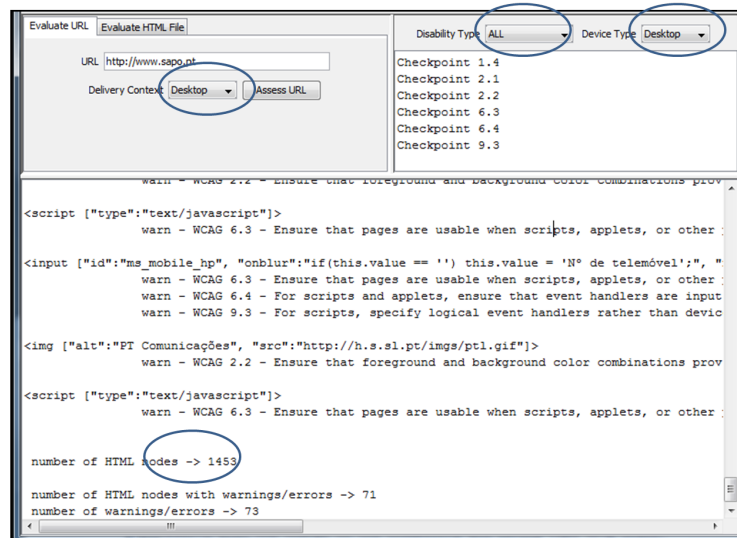


Figure 6.7: Default representation accessibility evaluation

In figure 6.8 we can see the assessment of the same URL, but now of the mobile representation and considering only the mobile adequacy guidelines. Notice the change

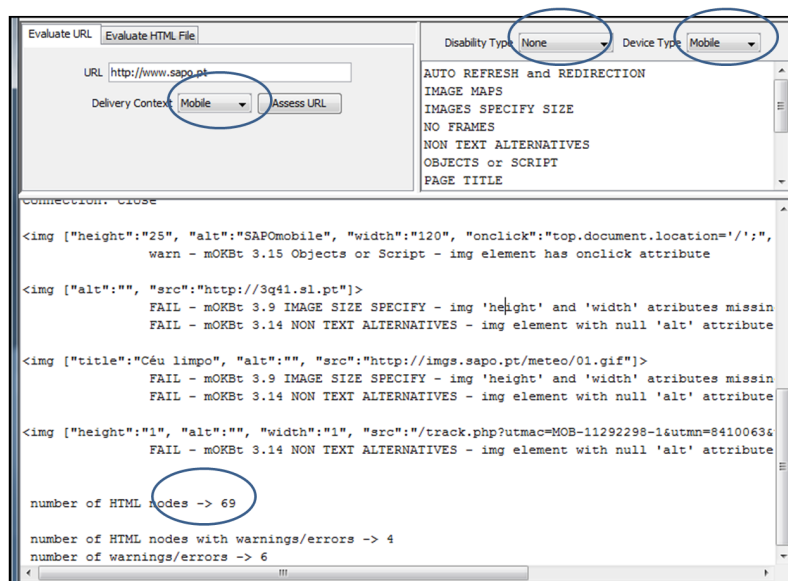


Figure 6.8: Mobile representation mobile adequacy evaluation

on the selection of the mobile delivery context, on the upper left panel of the figure. By choosing the mobile delivery context, the HTTP request headers will be set in order to negotiate with the Web server a content representation that satisfies the mobile OK requirements. Observe also the change on the Disability and Device Type drop boxes.

As we can see from figure 6.8, the SAPO portal improves the mobile user experience by supporting an adapted representation of the Web resource much more adequate to be

delivered to a mobile device. The HTML document received from a mobile access as only 69 HTML document nodes instead of the previous shown 1453 nodes. Using MWATT, the user is able to see the HTML source code of the received response and compare the different representations. This capability of accessing the different representation of an URL and evaluate these different Web contents is, as far as we know, not supported by other tools.

Looking at the results we can see that this 69 HTML node content has potential problems in 4 of them. For instance an image has an associated onclick attribute that many mobile devices might not support (first report of the results panel on figure 6.8). There are three images with null alt text attributes that will impact the rendering of the content on a mobile device, for example, whenever there is a narrow bandwidth connection or the mobile device browser is configured to avoid images' downloading. Also, there are two images whose size is not specified as should be for a mobile device rendering.

The Web content developer can easily edit and solve the problems found in this Web content, which although prepared to be accessed by a mobile device, still has some issues.

One of the key results of MWATT is the ability to combine the evaluation of mobile devices adequacy and accessibility. This feature is not supported by any tool we know. In figure 6.9 we can see the results of the accessibility evaluation of the mobile representation. Notice the change on the Disability Type selection.

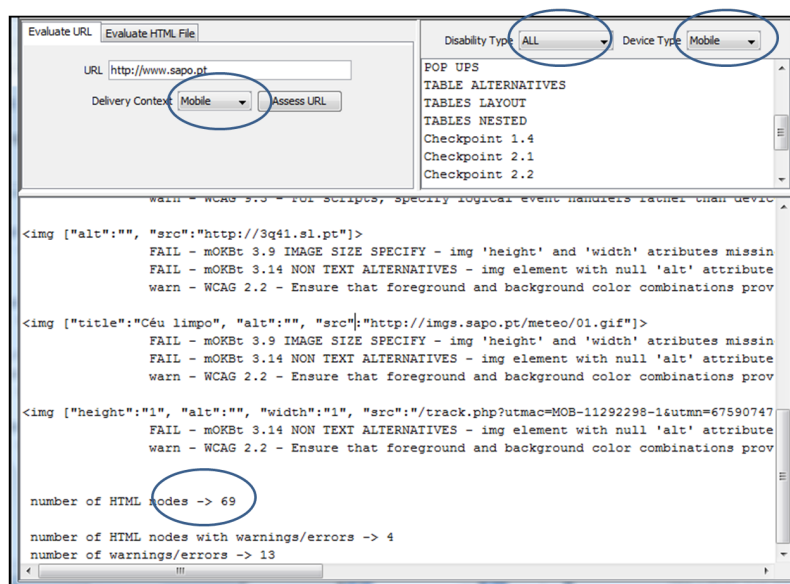


Figure 6.9: Mobile accessibility evaluation

On the results panel we can see that the same 69 HTML nodes now have 4 nodes with potential problems, but the total number of warnings and errors is 13, which means that some nodes have issues that will impact in more than one circumstances. For example, the last `` presented fails the mobile OK basic test for the existence of significant alternative text and also alerts for the contrast verification of the foreground and background

color.

MWAAT also allows the evaluation of the Web content regarding the aspects that are relevant for a specific type of disability as we can see from the following figures. Again this is another important result obtained with MWAAT.

Figure 6.10 shows the results for the mobile accessibility evaluation of SAPO mobile representation for users with blindness impairment. Figure 6.11 presents the results for

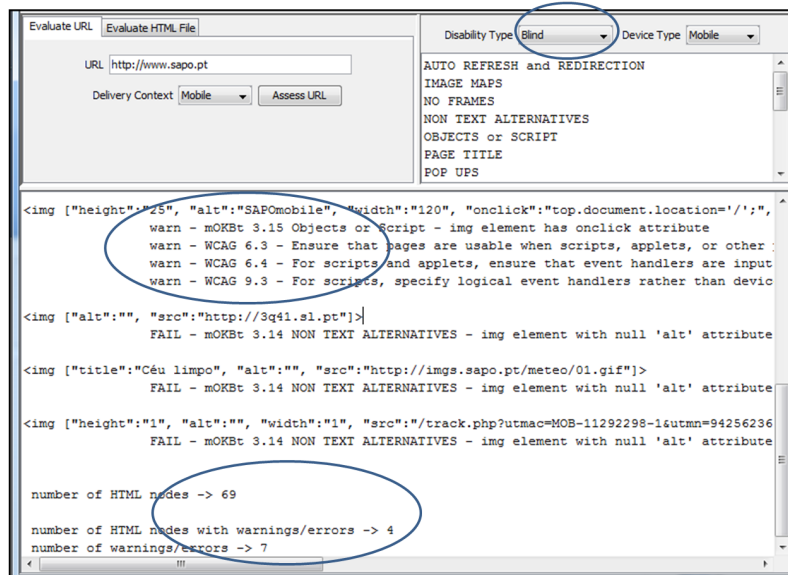


Figure 6.10: Mobile accessibility evaluation for blind disability

the mobile accessibility evaluation of SAPO mobile representation for users with deaf disability type. We can see for the blind disability that the document with 69 HTML nodes

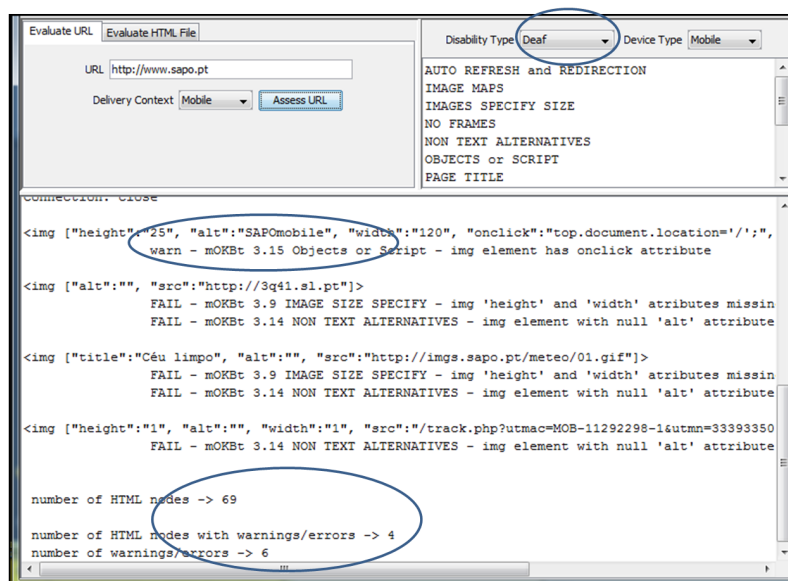


Figure 6.11: Mobile accessibility evaluation for deaf disability

presents 4 with 7 potential problems. For the deaf case, the same HTML representation presents only 6 potential issues.

On figures 6.10 and 6.11 we can see some details on the differences between the two profiles.

The `` tag includes an `onclick` attribute that might raise problems when the access is made from mobile devices whose hardware characteristics do not support click actions. This is independent of the disability type. However, for blind users, it may raise additional issues regarding scripts, applets, or other programmatic objects. These may be turned off or not supported by the assistive technology used, thus turning the page unusable. From figures 6.10 and 6.11, the developer is also informed of two other potential barriers for users with the deaf disability, that are not relevant for blind users, regarding the images size specification.

Figure 6.12 shows the results regarding a color blind disability, for the same Web content. Ten issues were signaled. They respect to the alternative texts' existence and to

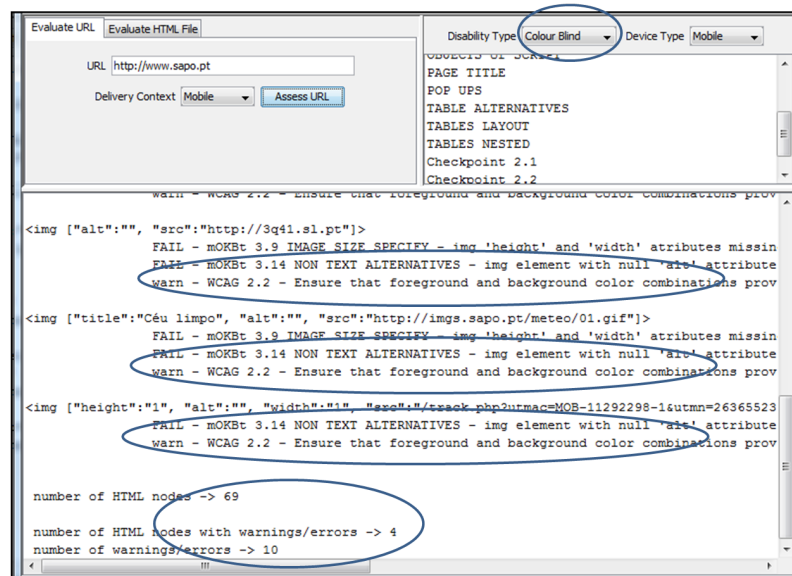


Figure 6.12: Mobile accessibility evaluation for color blind disability

the foreground and background color combinations that should provide sufficient contrast when viewed by someone having color deficits.

Regarding motor impaired, the assessments' results is shown in figure 6.13. Here on 4 nodes 9 issues were identified. From figures 6.10, 6.11, 6.12 and figure 6.13, we see that the same Web contents might raise different potential barriers depending on the specificities of the disability.

The remainder evaluation scenarios, as referred in beginning of the chapter, offer specific results. However, the exhaustive presentation of the corresponding screen captures will be too extensive. The results will summarized and discussed briefly in section 6.3.

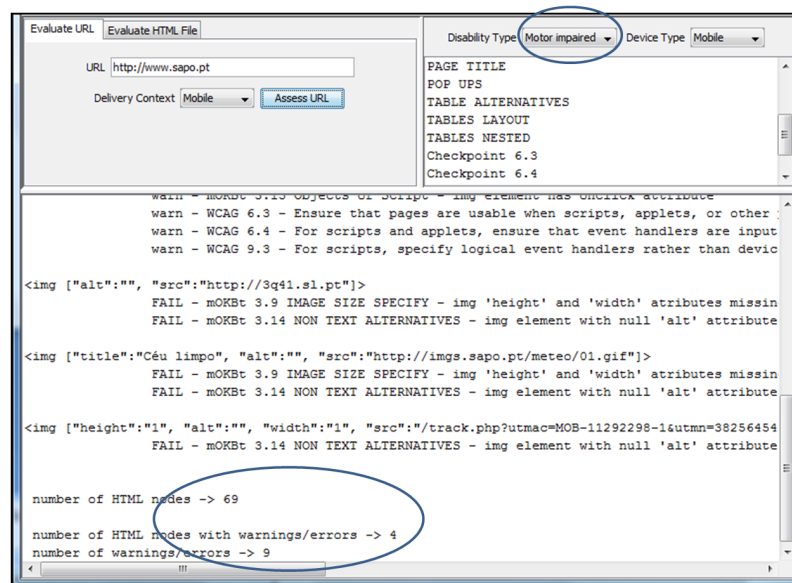


Figure 6.13: Mobile accessibility evaluation for motor impaired disability

6.2.2 Web Magazine

Engadget (at <http://www.engadget.com/>) is a Web magazine with daily coverage of everything new in gadgets and consumer electronics. It was launched in March of 2004 in partnership with the Weblogs, Inc. Network (WIN). Figure 6.14 Engadget home page

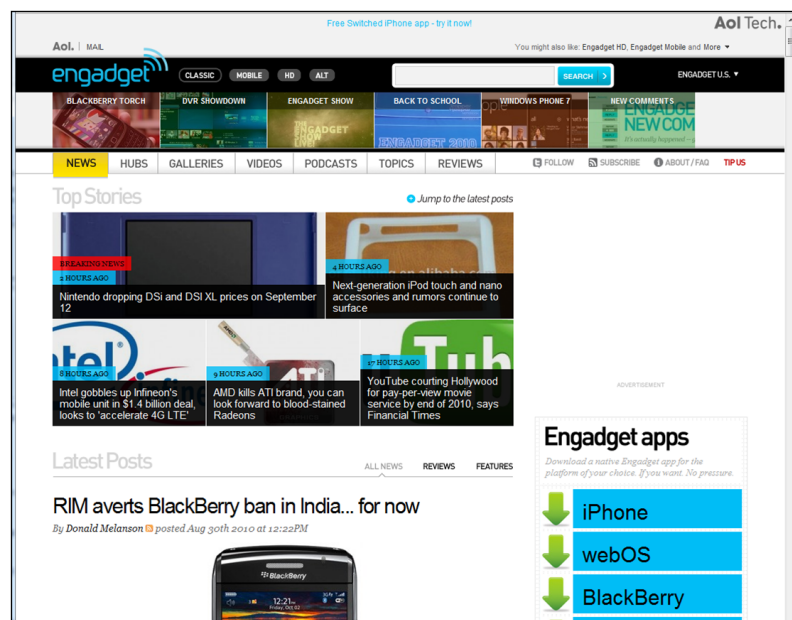


Figure 6.14: Engadget Home Page rendered on a desktop browser

also provides a mobile specific representation and presents an interesting challenge in two fronts: first because it is always full of pictures of the latest gadgets and key players; secondly because being a geek oriented magazine raises expectations regarding its adequacy

to mobile devices.

6.2.3 Web Finance Portal

Yahoo! Finance (at <http://finance.yahoo.com/>) offers information on stock quotes, up to date news, portfolio management resources, international market data, message boards and other financial related data. The critical financial information is highly dynamic, fre-

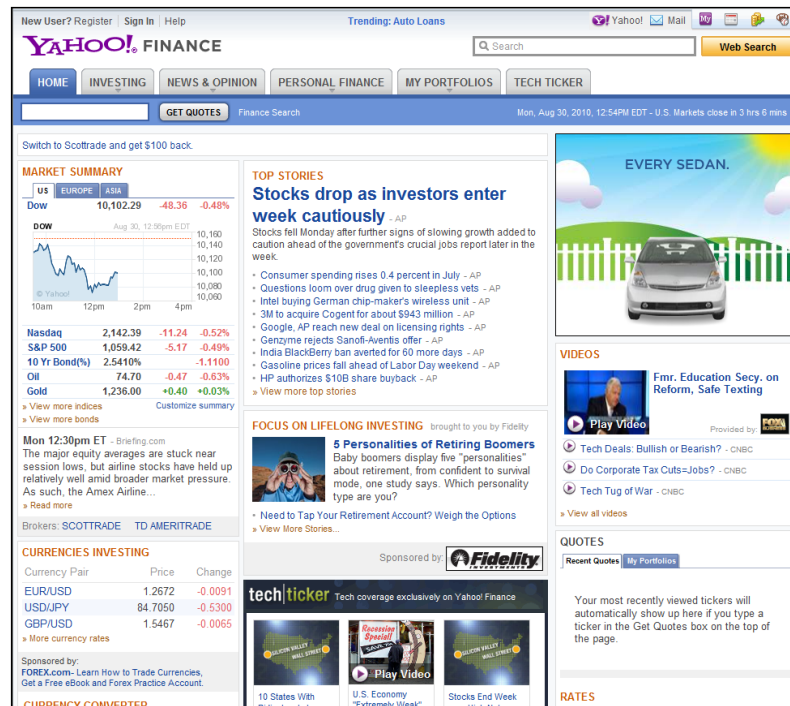


Figure 6.15: Yahoo Finance home page rendered on a desktop browser

quently accessed and needs to be always available on usage situations that should require mobile access. Again it also offers a specific mobile representation.

6.3 Aggregated Results

The tables bellow show the results for the three use cases presented above, covering the evaluation scenarios presented at the beginning of the chapter. For simplicity reasons, the tables for each case study are organized as follows.

- The first one reflects the accessibility evaluation for the default representation according to the scenarios presented on this chapter's beginning, namely, the assessment for no specific disability, for blind, deaf, color blind and motor impaired disability types, in this order.
- The second table reflects the mobile accessibility evaluation for the mobile representation. On this second table, the first row (Disability: NONE) depicts the mobile

assessment not considering accessibility issues whereas the next ones follow the abovementioned organization.

- The third table reflects the mobile accessibility evaluation for the default representation, using the same ordering of the second one.

The fourth percentage column represents the ratio between the number of nodes with warnings and the number nodes. The last column represents the ratio between the number warnings and the number nodes. Warn/Err signifies that fails and warnings were counted together.

Default Representation - Accessibility					
Disability	Nodes	Nodes with Warn/Err	%	Warn/Err	%
ALL	1453	71	5	73	5
Blind	1453	6	0	8	1
Deaf	1453	0	0	0	0
Color Blind	1453	65	4	65	4
Motor Impaired	1453	6	0	8	1

Mobile Representation - Mobile Accessibility					
Disability	Nodes	Nodes with Warn/Err	%	Warn/Err	%
NONE	69	4	6	6	9
ALL	69	4	6	13	19
Blind	69	4	6	7	10
Deaf	69	4	6	6	9
Color Blind	69	4	6	10	14
Motor Impaired	69	4	6	9	13

Default Representation - Mobile Accessibility					
Disability	Nodes	Nodes with Warn/Err	%	Warn/Err	%
NONE	1453	137	9	142	10
ALL	1453	137	9	215	15
Blind	1453	77	5	85	6
Deaf	1453	137	9	142	10
Color Blind	1453	137	9	207	14
Motor Impaired	1453	137	9	150	10

Table 6.1: Case Study - Web Portal Accessibility Evaluation

We see that on the first case study, although they support a mobile OK representation, the Web content even in that representation still maintains several non adequate issues such as onclick events with associated Java scripts that most mobile devices do not support, and more astonishing they maintain on several `` tag, the absence of a non null alt text and the absence of the height and/or width specification with the resulting rendering issues therefore derived.

Default Representation - Accessibility					
Disability	Nodes	Nodes with Warn/Err	%	Warn/Err	%
ALL	2583	204	8	210	8
Blind	2583	41	2	47	2
Deaf	2583	0	0	0	0
Color Blind	2583	163	6	163	6
Motor Impaired	2583	41	2	47	2

Mobile Representation - Mobile Accessibility					
Disability	Nodes	Nodes with Warn/Err	%	Warn/Err	%
NONE	71	1	1	1	1
ALL	71	1	1	2	3
Blind	71	0	0	0	0
Deaf	71	1	1	1	1
Color Blind	71	1	1	2	3
Motor Impaired	71	1	1	1	1

Default Representation - Mobile Accessibility					
Disability	Nodes	Nodes with Warn/Err	%	Warn/Err	%
NONE	2587	226	9	378	15
ALL	2587	227	9	589	23
Blind	2587	203	8	268	10
Deaf	2587	226	9	378	15
Color Blind	2587	227	9	542	21
Motor Impaired	2587	226	9	425	16

Table 6.2: Case Study - Web Magazine Accessibility Evaluation

To start with, consider the accessibility dimension. The first table of each case study shows some figures that raise interesting questions. Primarily, it is clear that most of the specific disabilities have much less issues than the general case, since each disability relevant set of guidelines is a subset of the available tests. Furthermore, a deeper analysis of the evaluation results showed that even when the numbers are similar between disabilities, the actual raised issues generally correspond to different guidelines.

This reinforces the decision of having a specific disability testing option, since for example, for the deaf case the site is completely accessible. At the other end, the color blind case, that has a significant number of issues, can be easily explained by the fact that most of them are warnings regarding the contrast of background and foreground colors. In fact, the actual contrast is not tested and might be correct. Still on the same dimension, it is worth notice that the SAPO and Yahoo financial case studies show better results than the Web Magazine one.

Looking now at the mobility dimension, we should focus the attention on the first line of the second and third tables of each case study. Here it is clear that the mobile repre-

Default Representation - Accessibility					
Disability	Nodes	Nodes with Warn/Err	%	Warn/Err	%
ALL	1424	35	2	35	2
Blind	1424	9	1	9	1
Deaf	1424	0	0	0	0
Color Blind	1424	26	2	26	2
Motor Impaired	1424	9	1	9	1

Mobile Representation - Mobile Accessibility					
Disability	Nodes	Nodes with Warn/Err	%	Warn/Err	%
NONE	252	4	2	7	3
ALL	252	4	2	11	4
Blind	252	3	1	3	1
Deaf	252	4	2	7	3
Color Blind	252	4	2	11	4
Motor Impaired	252	4	2	7	3

Default Representation - Mobile Accessibility					
Disability	Nodes	Nodes with Warn/Err	%	Warn/Err	%
NONE	1424	46	3	46	3
ALL	1424	46	3	81	6
Blind	1424	20	1	29	2
Deaf	1424	46	3	46	3
Color Blind	1424	46	3	72	5
Motor Impaired	1424	46	3	55	4

Table 6.3: Case Study - Web Finance Portal Accessibility Evaluation

sentation (second table) presents a much smaller page size than the default representation (third table) in all case studies. This occurs since all sites have a mobile specific content representation, which usually offer a much simplified version of the site. For sites without this feature the results of the second and third table would be the same.

Regarding the three cases and the percentage of issues raised, it is noticeable that the Web Magazine shows the best improvement rates (9% to 1% in nodes and 15% to 1% in warnings), from the default to the mobile representations. The Yahoo Financial shows no improvement, but the site is already quite good, in its default format (third table). Finally the SAPO case shows minimal improvement (9% to 6% and 10% to 9%). A deeper regard at the HTML content and the evaluation results shows the existence of an advertisement using javascript code, not supported by many mobile phones, and some `` tags with unspecified “alt” attributes and/or image size. Interestingly some of the “alt” attributes are there, but empty. Some of these cases can be seen in figures 6.8 to 6.13.

Considering the first two lines of the second table of each case study one can observe that the number of warnings augments as expected from the mobile adequacy case to

the mobile accessibility case (SAPO: 6 to 13; Magazine: 1 to 2; Yahoo: 7 to 11). This reinforces the notion that assessing mobile adequacy is not the same than assessing mobile accessibility.

Comparing the same two initial rows with the ones of the third table, it is patent that, regarding absolute numbers, the gain in accessibility is enormous when comparing the mobile representation version with the default one, both in terms of number of nodes with warnings and warnings. Looking at percentages, that is also true for most case studies. The exception is the number of warnings in the SAPO's case (last column of the tables). Again, the explanation falls on the existence of the aforementioned advertisement.

Overall, the verified improvement in the accessibility of the mobile representations versus the default ones is in accordance with the well known overlapping of guidelines. Nevertheless, this overlapping is not complete, far from it, and sometimes the reasons and the corrective measures are different between the mobile adequacy issues and the accessibility ones.

Finally, it is worth comparing the rows of the second table (or of the third, which follows a similar pattern). In all case studies the differences the general accessibility evaluation and the specific disability ones are noticeable, as referred in the analysis of the first table. An interesting one is the blind disability evaluation scenario. Particularly in the Magazine case the mobile accessibility issues disappear. A more detailed analysis of the results shows that the issue found on the mobile adequacy (first row) that disappears in the blind assessment (third row) is referring the missing image size specification, which is considered irrelevant in this later one.

6.4 Summary

This chapter presents results of the application of the MWAAT tool to a series of Web pages. First an introductory example of MWATT's usage is described showing its application to the development of mobile accessible Web content. Afterwards, three URL case studies evaluations are presented. On the first one, some detailed results are described to illustrate the reporting capabilities of the tool to provide the information necessary to assess the quality of the sites. Finally, a set of aggregated results is analyzed, revealing the different aspects of the proposed methodology and showing some relevant evaluation peculiarities.

Chapter 7

Conclusions and Future Work

The use of mobile devices to access the Web is increasing due to the proliferation of mobile devices, telecommunication networks and the boost of devices capabilities. Today many people depend on their mobile and smart phones to accomplish various tasks, including work, entertainment, information consumption, communication or socialization. These gadgets are growingly becoming an intrinsic part of peoples lives.

On another strand, the Web allied with this dissemination of technology encompasses a larger community and in particular becomes an opportunity for use by people with special needs. In fact, the growing features of these devices permit its usage as a way to help people with impairments and most definitely are able to increase their access to information and services. It is therefore becoming an obligation, in moral, legal and commercial terms, to create mobile Web accessible contents.

This obligation, though, should not be taken lightly. It is easy to state, although less easy to accomplish, that a Web content should comply with all the restrictions of all the possible impairments. That could lead to over simplistic content, or to enormous amounts of work that most of the time is ignored by developers, either by ignorance, time or most often budget constraints. Sometimes even there might exist incompatible solutions. An interesting approach is then to support the assessment and understanding of the requirements for each different disability types.

This thesis contributed to this endeavor: find support to the development of accessible, mobile Web content by supporting its assessment for each impairment or to all of them. The contribution includes two main results: first, a methodology that comprehends mobile adequacy assessment and accessibility evaluation; secondly a proof of concept tool that supports the Web content development process according to that methodology.

The former, results from an effort to adequately integrate two sets of guidelines, MWBP and WCAG, with specific disability type selection. It is well known that these sets overlap, some conditions fully overlap, some only partially and other not at all. On top of that, we claim that when specific disabilities are addressed some MWBP might become irrelevant. The argument holds, for example, in a blind scenario when mobile

content is addressed. For one, an adequately customized mobile browser will avoid the loading of images which will make no sense for a blind user anyway. Then, when assessing the adequacy of some Web page for a mobile setting, the MWBP guidelines that consider image size become irrelevant. Overall we maintain that an adequate methodology for the evaluation of mobile accessibility for specific disabilities must go beyond a simple merging of guidelines.

The second contribution presented in this dissertation is a proof of concept tool, MWAAT, which fully addressed the basic concepts of the proposed methodology. As other available tools, it offers mechanism for WCAG assessment or MWBP evaluation in an independent manner. However, unlike any other, it permits the evaluation of Web content through an adequate combination of both guidelines sets, as a whole, or more importantly, for a specific disability. As such, it provides a powerful mechanism to Web content developers and most notably to consolidate the selective disability assessment approach.

This dissertation illustrates these capabilities with three representative case studies. The results herein shown reveal not only the assessment differences between desktop and mobile content, but especially the difference between these at the impairment level.

Finally, regarding the development perspective, MWAAT adopts a very flexible strand envisioning its future integration into the ACCESSIBLE project platform and its potential extension into other guideline sets and alternative scenarios. Overall, it can be said that the resulting work fulfills the requirements of the project where it was developed and most certainly the goals established for this thesis.

7.1 Future Work

MWAAT still have many aspects to evolve, the most important being the base set of guidelines it comprehends, the User Interface improvements and extensions, the normalization and recreation of its reporting components and its integration with the ACCESSIBLE selection engine.

7.1.1 Set of Available Tests

The extension of the guidelines sets, and more specifically the implemented tests, should follow a twofold path. First, the full extent of mobile OK tests that still might be implemented should be completed. This along with the full integration of WCAG derived tests, would enable the tool to extend from a proof of concept to a directly applicable instrument to be used by Web content developers.

Secondly, regarding the WCAG set, the evolution from WCAG 1.0 to WCAG 2.0 tests is crucial. The current integration allowed us showing our approach, but as a tool it would gain deeply on a move to version 2. Clearly, some WCAG 2.0 tests may evolve from the

WCAG 1.0 ones. Whenever possible we will reuse existing code. In situations where reuse is not possible or worthwhile we will implement from scratch.

7.1.2 User Interface

Different architectural approaches are considered regarding the ACCESSIBLE user interface namely an overall Web portal and a standalone assessment version. The Web user interface will be a portal where relevant users can use Web based services of the accessible components namely the mobile accessibility component, to extract useful information for accessibility guidelines and evaluation results. Web services from each of the evaluator tools will be provided in order to be accessed from this Web User interface portal.

Users will also be able to download the standalone versions of the evaluation tools such as MWAAT with its own stand alone interface.

7.1.3 Results Handler

ACCESSIBLE project provides an evaluation and reporting language (EARL) based reporting tool. This tool will receive input from the evaluation modules and will interact with the user interface both the Web and the stand alone versions. The outcome will be an EARL based report containing information as defined within the EARL specification, regarding the user, the Subject, the Test case, and the specific evaluation results. In our future work plan results will also be extended in order to feed the EARL report tool that is being prepared by our partners in the ACCESSIBLE project.

7.1.4 Selector Component and Inference Engine

To really overcome the gap between designers and developers knowledge on accessibility and mobile technology related issues, and the development of mobile adequate and accessible Web contents, a multilayer ontological knowledge resource framework is being developed as part of the Accessible project.

This framework comprehends generic and domain specific accessibility concepts and their mapping to the assessment scenarios. It incorporates the accessible harmonized methodology that includes the extended work referred in the current dissertation and aims to formalize conceptual information of:

- characteristics of users with disabilities and assisted devices
- accessibility standards and associated guidelines and checkpoints
- semantic verification rules through appropriate Semantic Web rule Language (SWRL) and SPARQL rules and queries to help describe user groups and personas characteristics and associated constraints and their association to accessibility checkpoints.

The Rules Inference engine is being developed in order to connect the assessment applications such as MWAAT to the ACCESSIBLE ontologies in order to use the stored knowledge concerning the relevant accessibility guidelines for each usage scenario. The rule inference engine will communicate with the Web application assessment tools such as MWAAT in order to provide the necessary inference knowledge such as the relation between accessibility guidelines, functional limitations and impairments to be presented to the user for each usage scenario. The inference engine integration with MWAAT is part of the planned future work. This integration will replace the current existing Selector component.

7.1.5 Further Validation of Methodology and Tool

Besides the abovementioned case studies it is important that a final version of the tool, implementing a complete concretization of the methodology is fully validated. Further research should aim at experimentally determining that the method is reliable captures accessibility and mobile adequacy caveats and assess if it is efficient regarding other evaluation approaches.

Within ACCESSIBLE, pilot use of MWAAT next version will be used by developers for real Web site development and evaluation. This assessment will involve the evaluation of the methodology and the tools from the developer's perspective (in terms of functionality, performance, reliability and usability as perceived by the users of the tools), the evaluation from the technical-implementation perspective, (in terms of performance, software modularity, maintainability, extendibility, and functionality as measured and perceived by technical evaluation teams).

Regarding end-user's perspective, elderly and disabled end-users will be invited to assess the improvement of the accessibility level of the applications that were tested using the tools. Feedback received will be analyzed and whenever relevant will be incorporated as planned changes in the proposed methodology and tools.

7.1.6 Refining Assessment Scenarios

Mobile Web stakeholders assume that with the evolution of mobile devices Web content development regarding only desktop browsing or the simple mobile delivery context does not cope with the Web users needs. The recent evolution on the smart phones arena with devices such as iPhone and BlackBerry or the Android operating system introduce capabilities that are far beyond a typical mobile phone. In view of that new categories of devices are clearly emerging, which impacts on the definition of delivery contexts and on the selection of the adequate guidelines sets to be applied during assessment.

No work has been done so far regarding other delivery contexts' specification beyond the mobile OK delivery one. Therefore a deep study of existing devices, browsers and

assistive technologies must be undertaken, classifying its variability and building a more thorough taxonomy. This study could even comprehend contexts of use that often preclude some of the devices features. Thus, studies have to be conducted with real usage situations in order to determine alternative usage situations, user needs and eventual accessibility and technology associated barriers.

On the other hand, this newly defined delivery contexts will most certainly define alternative evaluation scenarios. Existing guidelines should be further analyzed regarding its adequacy to these new contexts. Its relevance should be assessed, requiring changes in some of them or even the definition of new ones. Again, detailed studies need to be performed and documented explaining the reasons why or why not each guideline or success criteria is significant or eventually needs to be evaluated and changed.

Bibliography

- [1] Shadi Abou-Zara. *Web Accessibility Evaluation Tools: Overview*, March 2006. Last accessed September 1st, 2010.
- [2] CETH.and all. Dow - description of work (annex i -). Technical report, ACCESSIBLE, Grant Agreement N° 224145, May 2008.
- [3] Rogério Bandeira, Rui Lopes, and Luís Carriço. Towards mobile web accessibility evaluation. In *Proceedings of the ETAPS 2010 FOSS-AMA Workshop*, Paphos, Cyprus, March 2010. Springer.
- [4] S. Benjamin and all. D 2.1 - state of the art survey in accessibility research and market survey. Technical report, ACCESSIBLE, Grant Agreement N° 224145, September 2009.
- [5] BITV. *BITV: Barrierefreie Informationstechnik-Verordnung (Federal Ordinance on Barrier- Free Information Technology)*, April 2002. Last accessed on January 19th, 2010.
- [6] United States Access Board. *Section 508 of the Rehabilitation Act. United States Access Board*, December 2008. Last accessed on January 19th, 2010.
- [7] Giorgio Brajnik. Web accessibility testing: when the method is the culprit. In *ICCHP 2006, 10th International Conference on Computers Helping People with Special Needs*, Linz, Austria, July 2006. ICCHP 2006.
- [8] Giorgio Brajnik. *Barrier Walkthrough Heuristic evaluation guided by accessibility barriers*, 2009. Last accessed on August 31, 2010.
- [9] Ben Caldwell, Michael Cooper, Loretta Guarino Reid, and Loretta Guarino Reid. *Web Content Accessibility Guidelines 2.0*, December 2008. Last accessed on 21 May 2010.
- [10] Eleni Chalkia and all. D 3.1 - ham accessible harmonized methodology. Technical report, ACCESSIBLE, Grant Agreement N° 224145, September 2009.

- [11] Wendy Chisholm, Gregg Vanderheiden, and Ian Jacobs. *Web Content Accessibility Guidelines 1.0*, May 1999. Last accessed on 21 May 2010.
- [12] Wendy Chisholm, Gregg Vanderheiden, and Ian Jacobs. *Techniques for Web Content Accessibility Guidelines 1.0*, november 2000. Last accessed on August 31, 2010.
- [13] Alan Chuter and Yeliz Yesilada. *Relationship between Mobile Web Best Practices (MWBP) and Web Content Accessibility Guidelines (WCAG)*, July 2009. Last accessed on 24 May 2010.
- [14] DotMobi. *mobiReady*, 2010. Last accessed on 05 August 2010.
- [15] ERCIM, MIT, and Keio. *W3C mobileOK Checker, V1.4.1*, May 2010. Last accessed on 05 August 2010.
- [16] Shawn Lawton Henry. *Essential Components of Web Accessibility*, August 2005. Last accessed on 21 May 2010.
- [17] Shawn Lawton Henry, participants of the Education, and Outreach Working Group. *Introduction to Web Accessibility*, September 2005. Last accessed on 20th May 2002.
- [18] HiSoftware. *HiSoftware Compliance Sheriff Accessibility Module*, 2000. Last accessed on January 19th, 2010.
- [19] Ian Jacobs, Jon Gunderson, and Eric Hansen. *User Agent Accessibility Guidelines 1.0*, December 2002. Last accessed on August 31, 2010.
- [20] I. Jacobson. *Object Oriented Software Engineering*. Addison-Wesley Professional, 1992. 1st Edition.
- [21] P. Korn and all. D 2.2a - user needs and system requirements specification. Technical report, ACCESSIBLE, Grant Agreement N° 224145, April 2009.
- [22] Craig Larman. *Applying UML and Patterns*. Prentice Hall PTR, 1992. 2st Edition.
- [23] Mauro Nunes and Henrique O’Neil. *Fundamental de UML*. FCA - Editora Informática, Lda, Lisbon, Portugal, September 2001.
- [24] B. Nuseibeh and all. Requirements engineering: A roadmap. In *Proceedings of the Conference on the Future of Software Engineering, International Conference on Software Engineering*, pp.35-46, Limerick, Ireland, September 2000.
- [25] World Health Organisation. *International Classification of Functioning, Disability and Health*, 2009. Last accessed January 19th, 2010.

- [26] Sean Owen and Jo Rabin. *W3C mobileOK Basic Tests 1.0*, November 2008. Last accessed on 1 September 2010.
- [27] Roger S. Pressman. *Software Engineering, A Practitioner's Approach*. McGraw Hill, 2005. Sixth Edition, International Edition.
- [28] Jo Rabin and Charles McCathieNevile. *Mobile Web Best Practices 1.0*, July 2008. Last accessed on 24 May 2010.
- [29] T.A.W. *T.A.W mobileOK Basic Checker*, 2010. Last accessed on 05 August 2010.
- [30] Jim Thatcher and Michael R. Burks. *Web Accessibility Web Standards and Regulatory Compliance*. friends of Ed an Apress Company, USA1, 2006.
- [31] Justin Thorp, Shawn Lawton Henry, participants of the Web Accessibility Initiative Education, and Outreach Working Group. *Web Content Accessibility and Mobile Web*, October 2008. Last accessed on 21 May 2010.
- [32] Jutta Treviranus, Charles McCathieNevile, Ian Jacobs, and Jan Richards. *Authoring Tool Accessibility Guidelines 1.0*, February 2002. Last accessed on August 31, 2010.
- [33] Dimitrios Tzovaras and all. D 3.3 - accessible system architecture specification. Technical report, ACCESSIBLE, Grant Agreement N° 224145, August 2009.
- [34] Usablenet. *Lift Machine*, 2009. Last accessed on January 19th, 2010.
- [35] Cynthia D. Waddell. Overview of law and guidelines. In *Web Accessibility Web Standards and Regulatory Compliance*, New York, NY, USA, 2006. Springer.
- [36] WebAIM. *WAVE Web Accessibility Evaluation Tool*, 2009. Last accessed on January 19th, 2010.
- [37] Dierk Konig with Andrew Glover, Paul King, Guillaume LAforge, and Jon Skeet. *Groovy in Action*. Manning Publications Co, 2007.
- [38] Yeliz Yesilada, Tianyi Chen, and Simon Harper. *Mobile Web Barriers for the Barrier Walkthrough*, July 2008. Last accessed on January 19th, 2010.

